Transfer Learning

COMPSCI 760
2024 Semester 1

Olivier Graffeuille
ogra439@aucklanduni.ac.nz

Outline

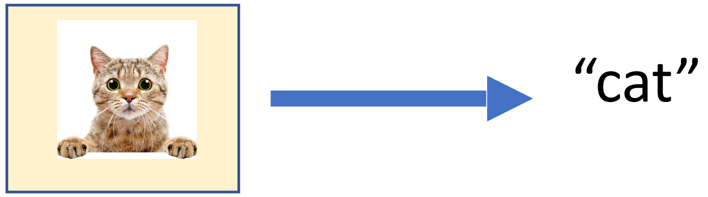
- Tasks & Domains
- What is Transfer Learning?
- Transfer Learning Methods
  - Instance Transfer
  - Feature Transfer
  - Parameter Transfer
- Multi-Task Learning

Outline

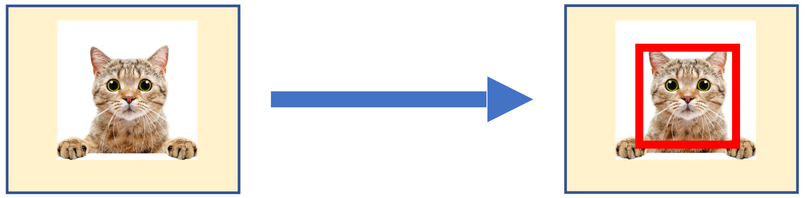
- ***Tasks & Domains***
- What is Transfer Learning?
- Transfer Learning Methods
    - Instance Transfer
    - Feature Transfer
    - Parameter Transfer
- Multi-Task Learning

Prediction in Machine Learning

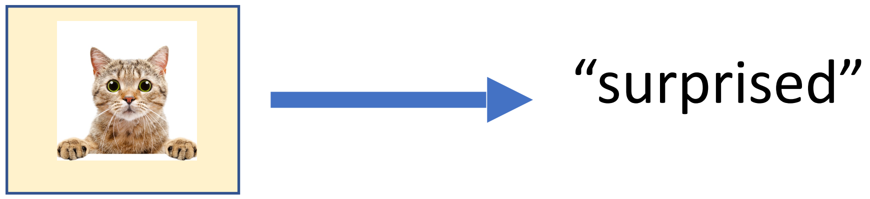
- Most ML techniques aim to predict a label from an input



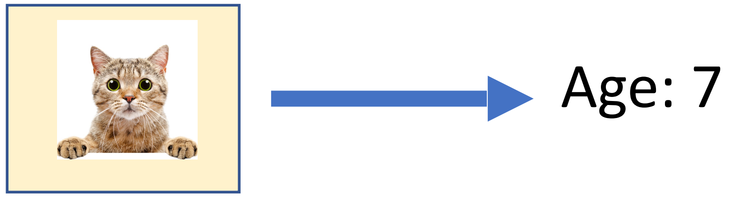
Classification



Object detection



Emotion Recognition



…cat age estimation?

Tasks (informally)

- A task is a function we are trying to learn
- i.e. what is the objective of the neural network?
    - Classify an image
    - Estimate the age of cat images
    - Detect cancer in 3D MRI scans
    - Speech to text

Tasks (formally)

- **Task:**

  A label space and a function.

  $$T = \{\mathcal{Y}, f(.)\}$$

  Where $\mathcal{Y}$ represents the label space
  e.g. {0, 1} for a binary classification problem, or any 19x19x16 dimensional matrix for YOLO

  And f(.) is a function that maps from $\mathcal{X}$ -> $\mathcal{Y}$ (we're trying to learn this function)
  e.g. a model that predicts the most likely class/object position, given an input

  Note:
  Discriminative models learn a function P(y | X)
  Generative models model P(X, y) instead

Domains (informally)

- The possible inputs to our model
- i.e. what kind of data do we want our model to handle?
  - Any image
  - Images of leaves of trees
  - Hand-drawn digits
  - Audio of people speaking Arabic
  - Medical records

Domains (formally)

- **Domain:**

  A feature space and marginal probability density function. $D = \{\mathcal{X}, p(X)\}$

  Where $\mathcal{X}$ represents the input space

  e.g. $\mathbb{R}$ *for a single variable model,* $\mathbb{R}^{28} x \mathbb{R}^{28}$ *for MNIST*

  And P(X) represents the marginal distribution of X

  e.g. a normal distribution for height, or a very complicated intractable distribution for MNIST

Classes, Domains and Tasks

- **Class:**

  A distinct category or group that a data point is labelled as belonging to.
  E.g. "Cat", "Dog", "Happy", "Popular", "Correct"

- **Domain:**

  $$D = \{\mathcal{X}, p(X)\}$$

  A feature space and marginal probability density function.
  E.g. Sunny photos of pets, painting of pets, audio recording of pet.

- **Task:**

  $$T = \{\mathcal{Y}, p(y|X)\}$$

  A label space and conditional probability density function.
  E.g. Does the image contain a dog or a cat? How many dogs are in this image?

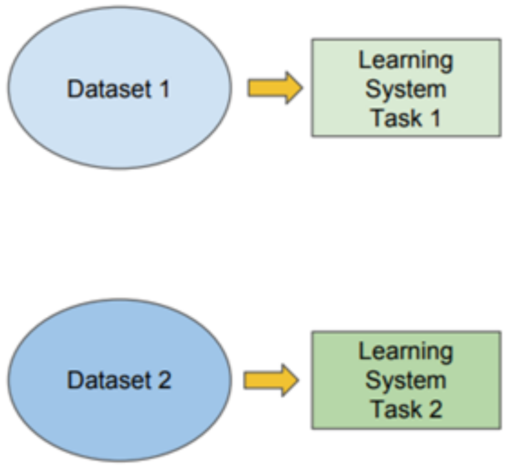
Outline

- Tasks & Domains
- ***What is Transfer Learning?***
- Transfer Learning Methods
    - Instance Transfer
    - Feature Transfer
    - Parameter Transfer
- Multi-Task Learning

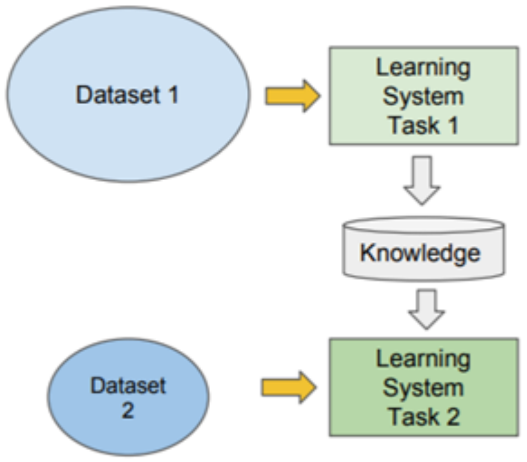
Transfer Learning (informally)

Traditional ML vs Transfer Learning

- Isolated, single task learning:
  - Knowledge is not retained or accumulated. Learning is performed w.o. considering past learned knowledge in other tasks



- Learning of a new tasks relies on the previous learned tasks:
  - Learning process can be faster, more accurate and/or need less training data

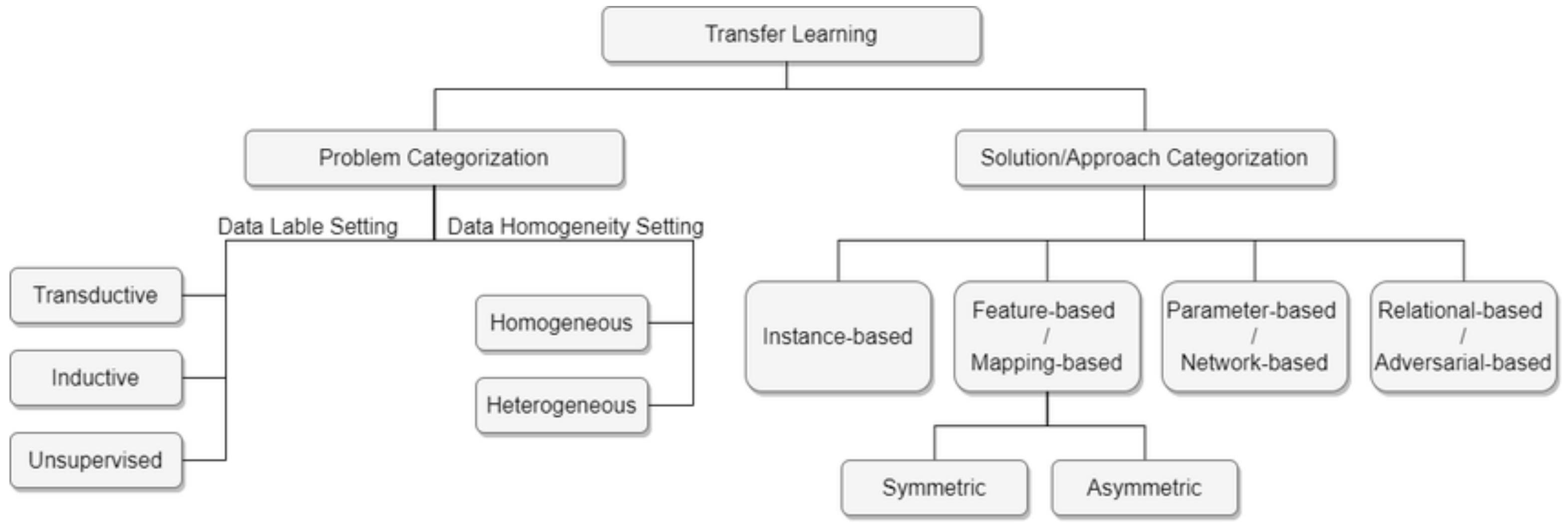


https://www.v7labs.com/blog/transfer-learning-guide

Transfer Learning (formally)

- Given a source domain D_S with corresponding learning task T_S and a target domain D_T with corresponding learning task T_T
- Transfer Learning (TL) is the process of improving the target predictive function f_T by using the related information from D_S and/or T_S, where $D_S \neq D_T$ or $T_S \neq T_T$
  - If $D_S = D_T$ and $T_S = T_T$ then the two tasks are just identical
- Source task generally has much more data than the target task

Transfer Learning Taxonomy



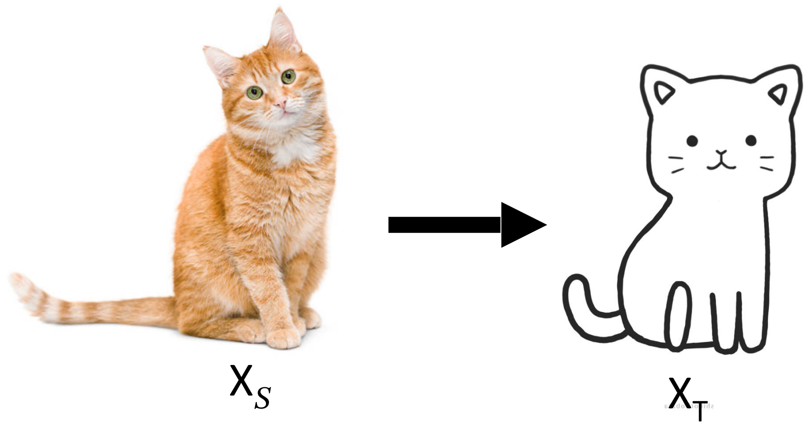
Change in domain

- $D_S \neq D_T$
  - $\mathcal{X}_S \neq \mathcal{X}_T$

    Heterogeneous TL
    Different "kinds" of input
    e.g. Transferring from one kind of sensor to another

  - $\mathcal{X}_S = \mathcal{X}_T$ but, $P(X_S) \neq P(X_T)$

    Domain adaptation
    Same "kind" of input but different distribution
    e.g. Transferring from real images to cartoon images



Change in task

- $T_S \neq T_T$
  - $\mathcal{Y}_S \neq \mathcal{Y}_T$

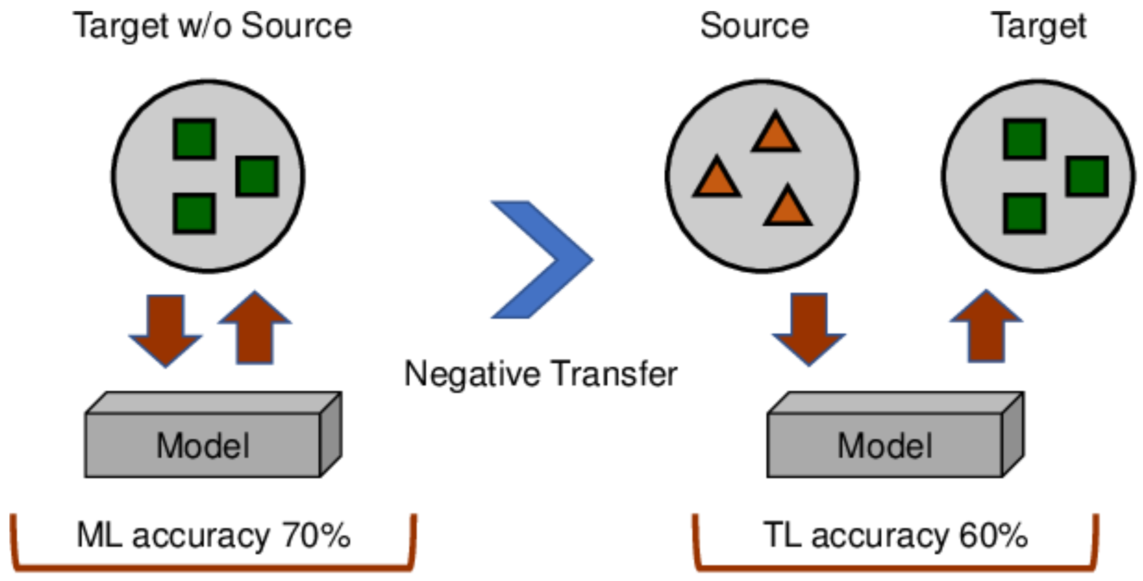
    Different "kinds" of labels
    e.g. Transferring image classification to image object detection

  - $f_S(.) \neq f_T(.)$ i.e. $P_S(y|x) \neq P_T(y|x)$

    Relationship between input and label is different
    e.g. Transferring from one language to another ("sale" means dirty in French)

Negative Transfer

- Sometimes transfer learning can reduce performance on target task
  - This is called **Negative Transfer**
- Transfer learning is not guaranteed to improve performance!

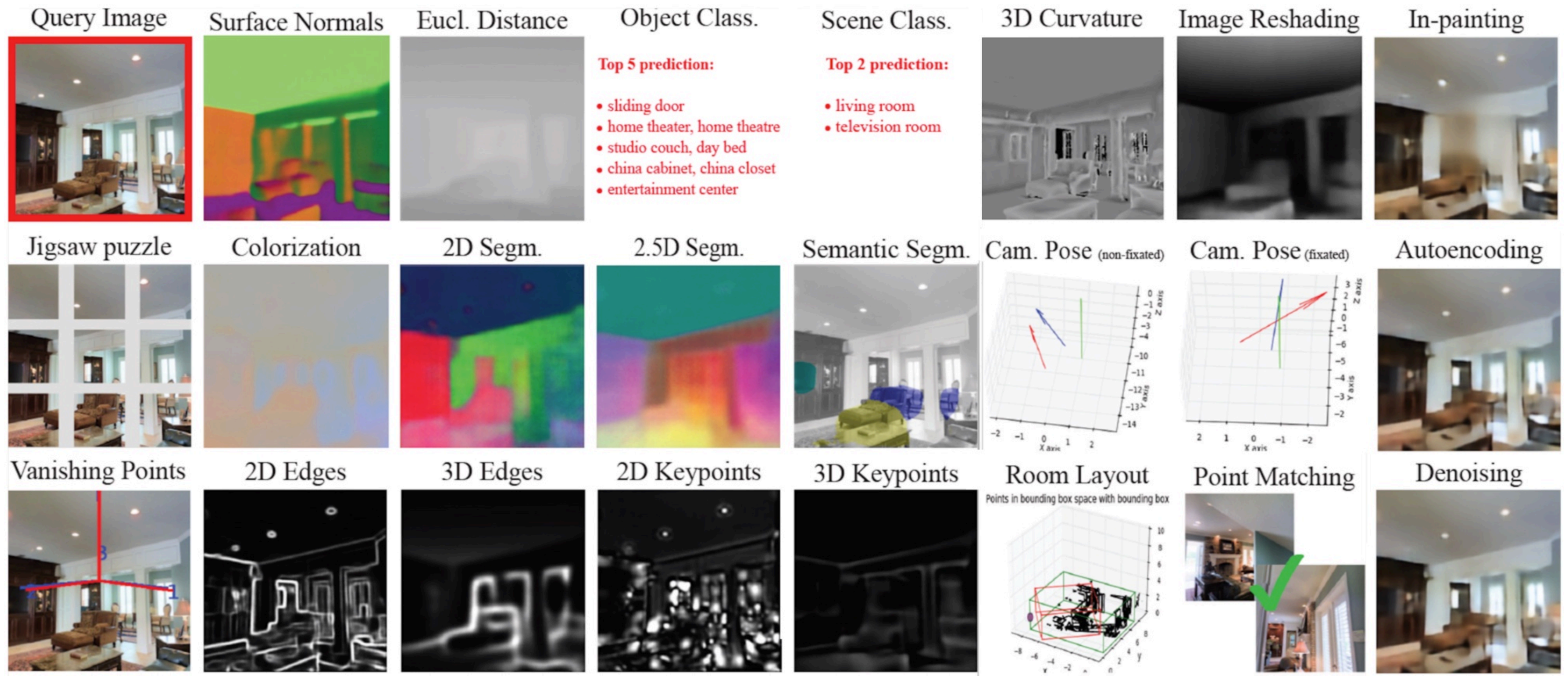


Negative Transfer

- Assumption in transfer learning: *tasks are related*
    - Transferring from Japanese alzheimers detection model to NZ alzheimers detection model? ✓
    - Transferring from image classification (imagenet) to car object detection? ✓
    - Transferring from an NLP task (e.g. Q&A) to object detection of cars? ✗
    - Transferring from different language for an NLP task (English -> Chinese)? ✓
    - Transferring from global climate model to ecology sustainability model?

Negative Transfer

- Demo of transfer of image tasks http://taskonomy.stanford.edu/transfers/

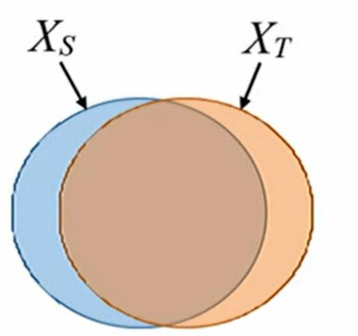


Outline

- Tasks & Domains
- What is Transfer Learning?
- ***Transfer Learning Methods***
    - ***Instance Transfer***
    - Feature Transfer
    - Parameter Transfer
- Multi-Task Learning

Instance Transfer

- Simplest solution:
  can we re-use some data samples from the source domain on the target domain directly?
- Re-weight each source data instance
- Assumptions:
  Source and target domain have same feature and label spaces
  Source and target domain have same task function
  Different marginal probability distributions



Instance Transfer

- Assumptions:

$\mathcal{Y}_S = \mathcal{Y}_T$
$P_S(y|x) = P_T(y|x)$
$\mathcal{X}_S = \mathcal{X}_T$
$P(X_S) \approx P(X_T)$

What transfer learning setting is this?

- Techniques: weight samples from source domain based on:
  - based on improvement in performance (TrAdaBoost)
  - based on relative probability of occurance $\frac{P(X_T)}{P(X_S)}$

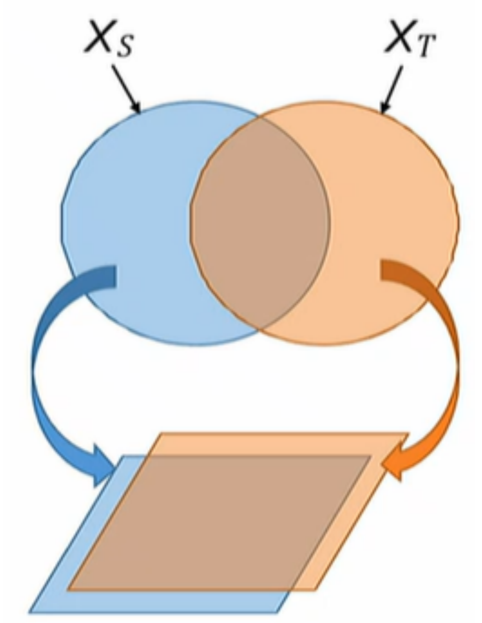
Outline

- Tasks & Domains
- What is Transfer Learning?
- Transfer Learning Methods
    - Instance Transfer
    - ***Feature Transfer***
    - Parameter Transfer
- Multi-Task Learning

Feature Transfer

- Find a common feature space for source and task domain $\mathcal{X}_S \neq \mathcal{X}_T$
  - *e.g. MRI scans and blood tests of patient*
    or: time series data + tabular data
    or: medical records from different hospitals which measure different variables

- This allows us to use the same classifier for the source and target model



Outline

- Tasks & Domains
- What is Transfer Learning?
- Transfer Learning Methods
    - Instance Transfer
    - Feature Transfer
    - ***Parameter Transfer***
- Multi-Task Learning

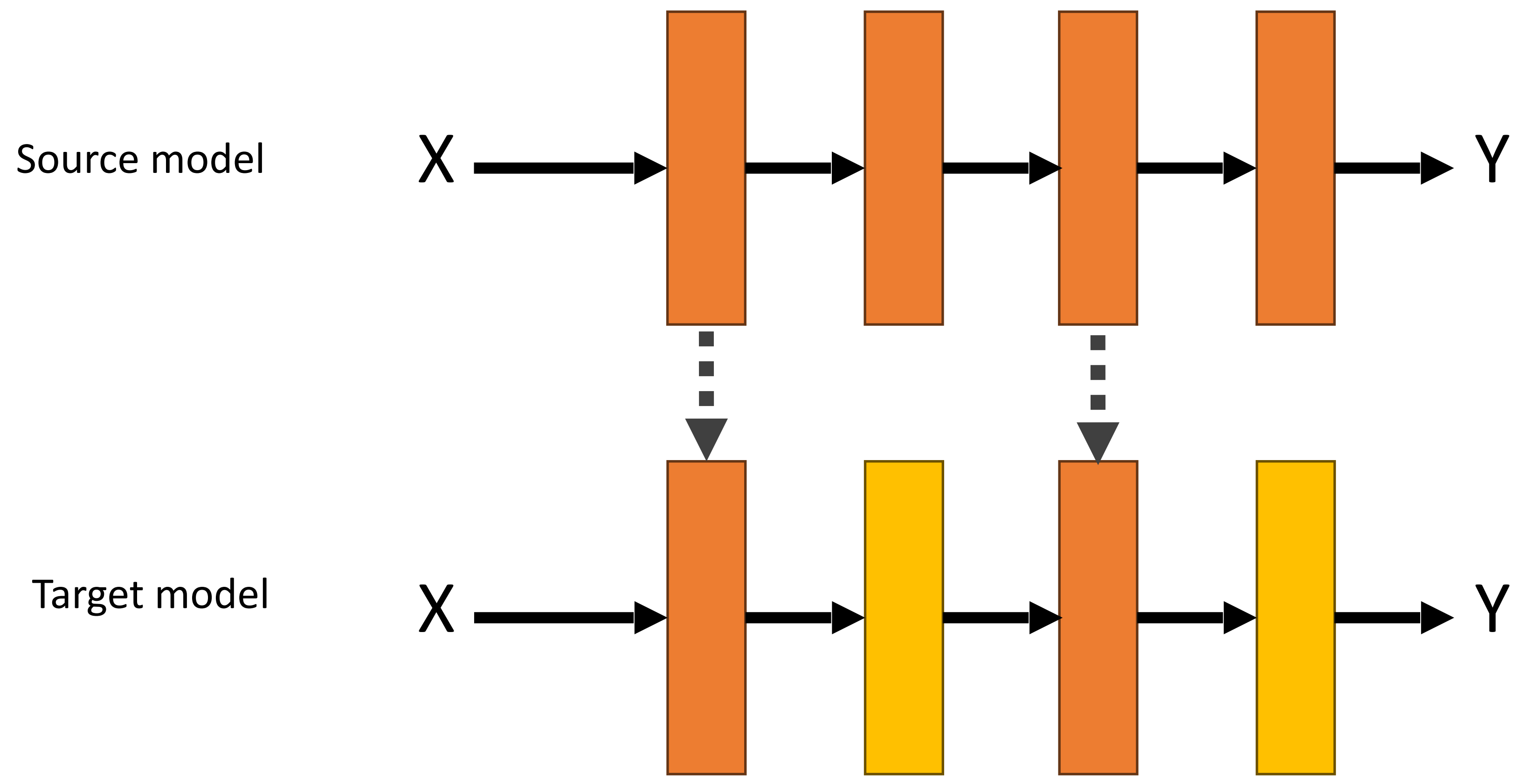
Parameter Transfer

- Transfer parameters from model trained on source dataset to target dataset
  - Intuition: models encode knowledge in their parameters
    So, we can transfer parameters to transfer knowledge
  - Transferring knowledge from source model to target model
- Decision trees can transfer decision nodes
- Neural networks can transfer model weights

Fine Tuning

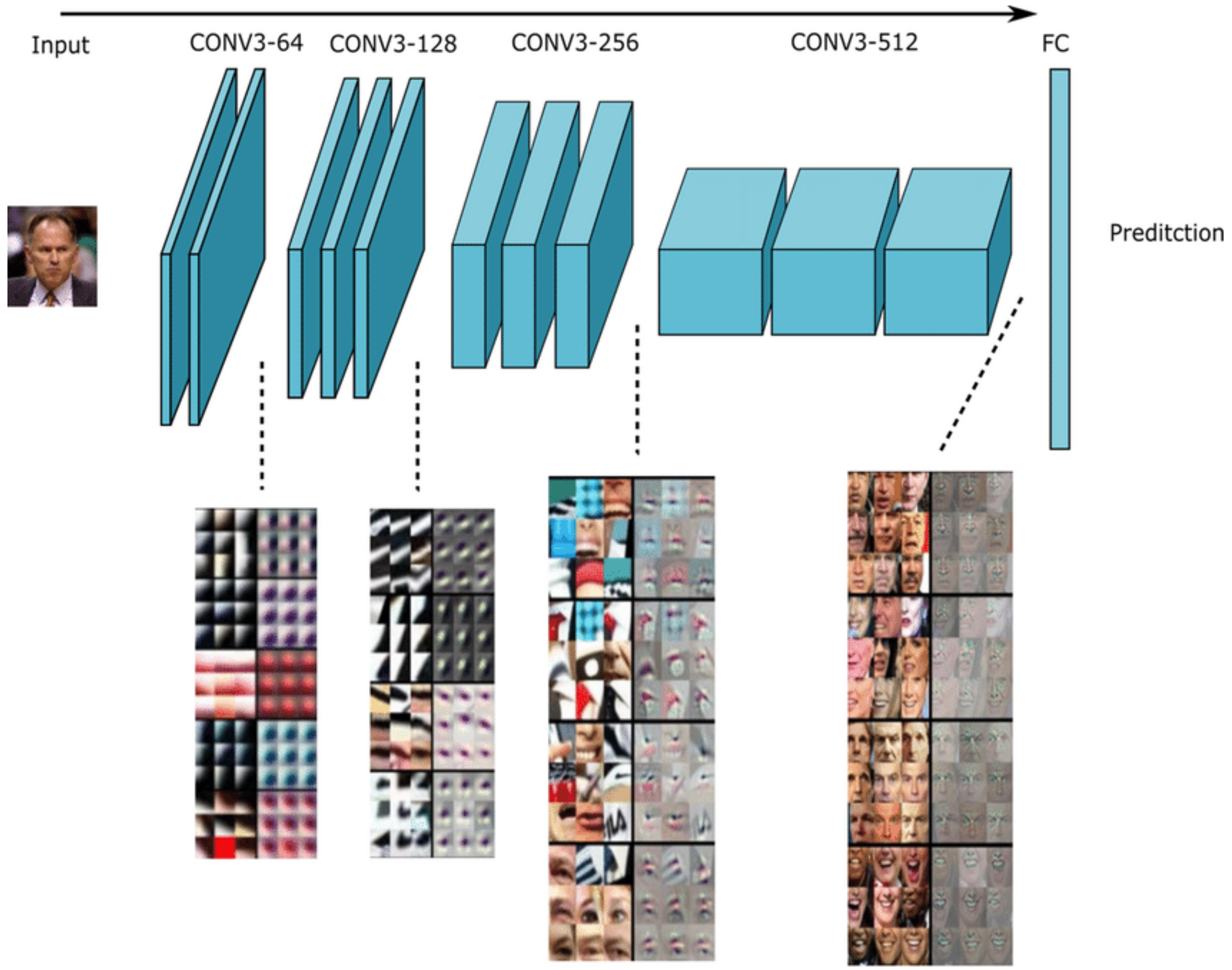
- Train model on source data, then fine-tune on target data
- Challenge: limited target data, careful to not overfit
- Techniques:
    - Limiting the number of training epochs
    - Use a smaller learning rate
    - Freeze some layers
    - Some combination (gradual unfreezing of select layers)

Which layers to transfer?

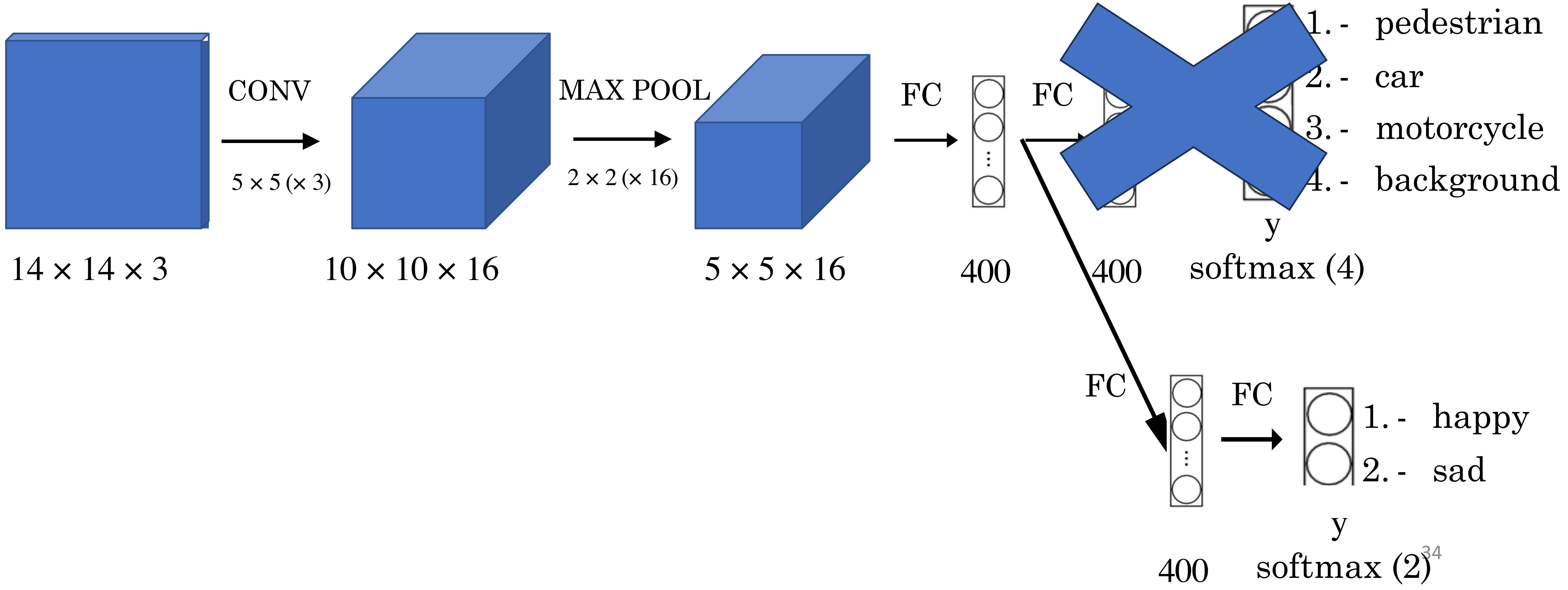


Which layers to transfer?

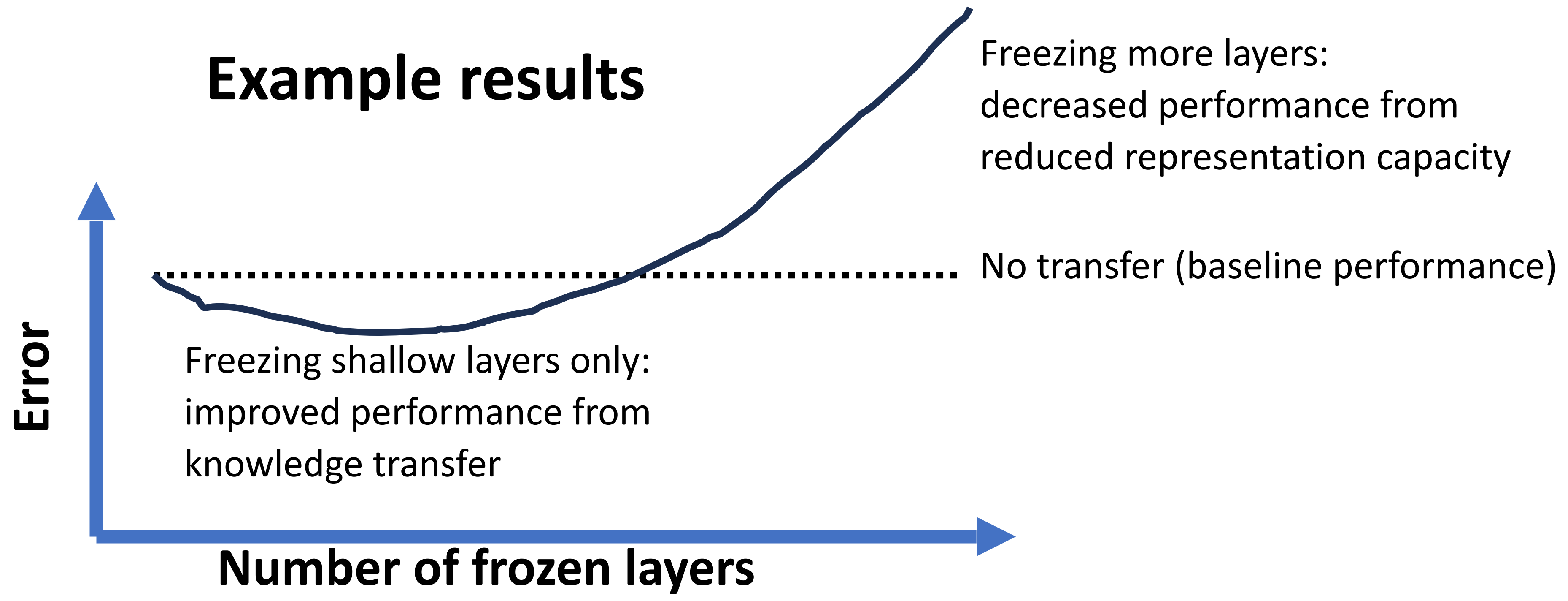
- It depends
- Computer vision:
  Usually transfer shallow (early) layers



Which layers to transfer?

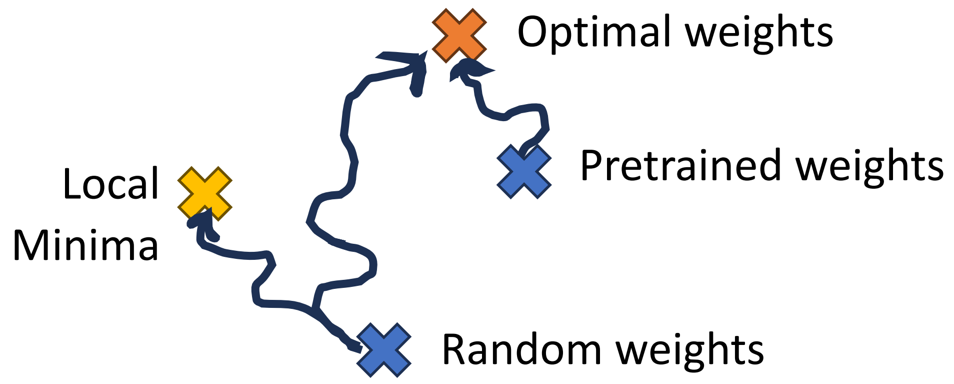


Which layers to transfer? Trade-off



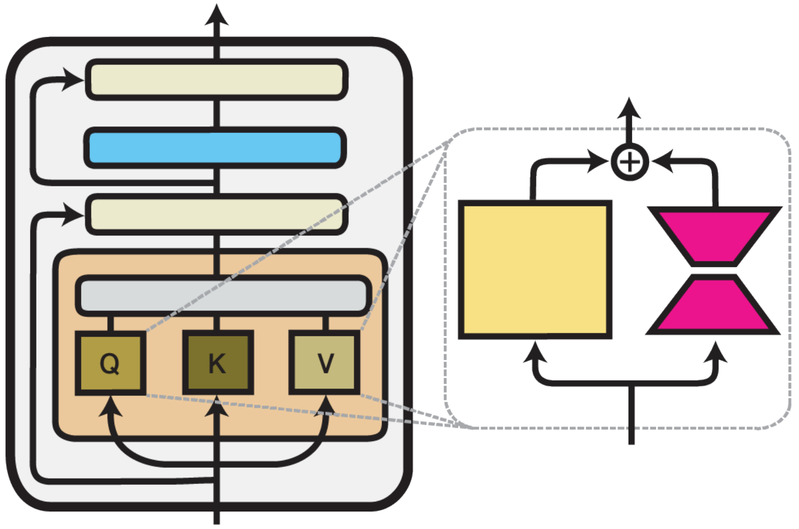
Pre-Trained Models

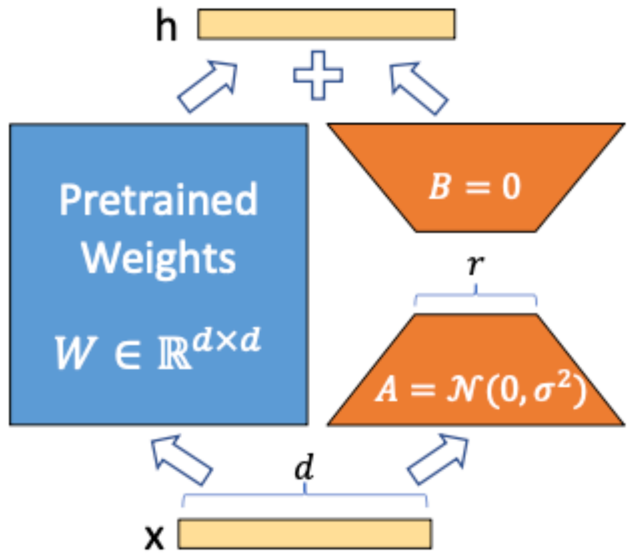
- When training neural networks, we can either:
  - Select a random initialisation for the network parameters (noise)
  - Select an initialisation from a pre-trained model
- Popular pre-trained models:
  - Computer vision
    - VGG16
    - ResNet-50
    - ViT
  - NLP
    - Bert
    - Llama 3



Adapters

- Small NN architecture which is inserted into pre-trained models
  - The pre-trained model is entirely frozen
  - The adapter is trained normally
- Example: LoRA
  - New weight matrix added (parallel to existing layers)
  - New weight matrix has many fewer parameters




h
Pretrained Weights
$W \in \mathbb{R}^{d \times d}$
$B = 0$
r
$A = \mathcal{N}(0, \sigma^2)$
d
x

[1] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, & Weizhu Chen. (2021). LoRA: Low-Rank Adaptation of Large Language Models. arXiv:2106.09685.

Transfer Learning Summary

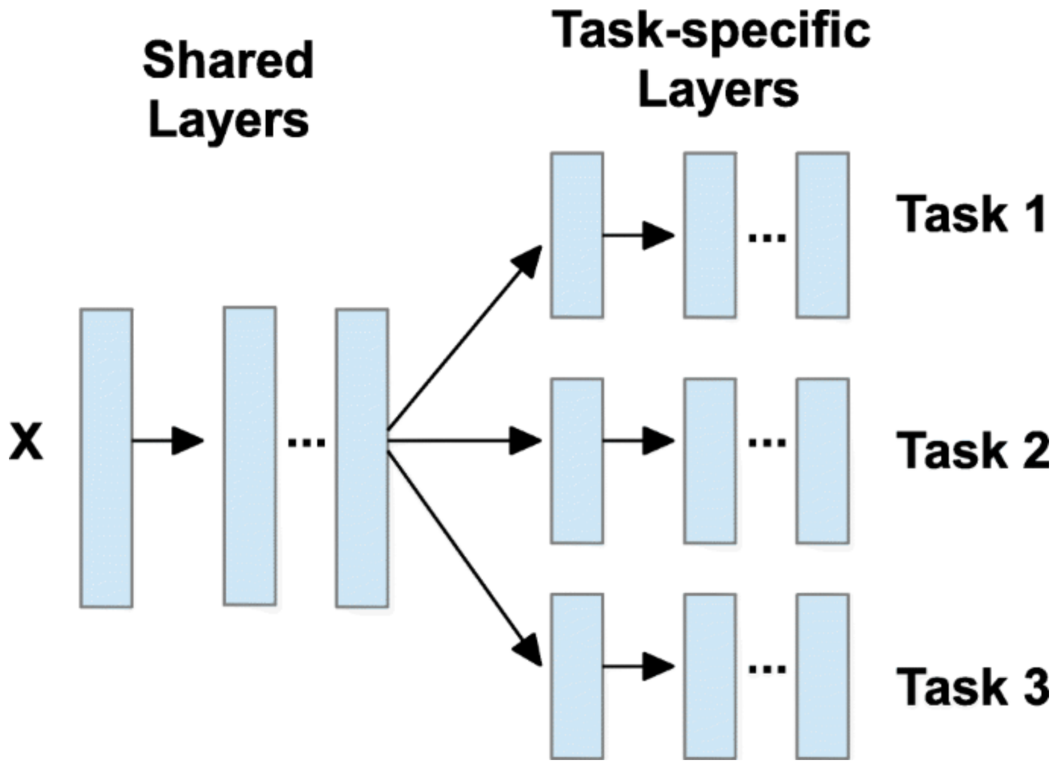
- Many pre-trained models can serve as the starting point for training. These pre-trained models give a more reliable architecture and save time and resources.
- Transfer learning:
  - used in scenarios where there is not enough data for training or when we want better results in a short amount of time.
  - involves selecting a source model similar to the target domain, adapting the source model to the target model before transferring the knowledge, and training the source model to achieve the target model.
- It is common to fine-tune the higher-level layers of the model while freezing the lower levels as the basic knowledge is the same that is transferred from the source task to the target task of the same domain.
- In tasks with a small amount of data, if the source model is too similar to the target model, there might be an issue of overfitting. To prevent the transfer learning model from overfitting, it is essential to tune the learning rate, freeze some layers from the source model, or add linear classifiers while training the target model can help avoid this issue.

Outline

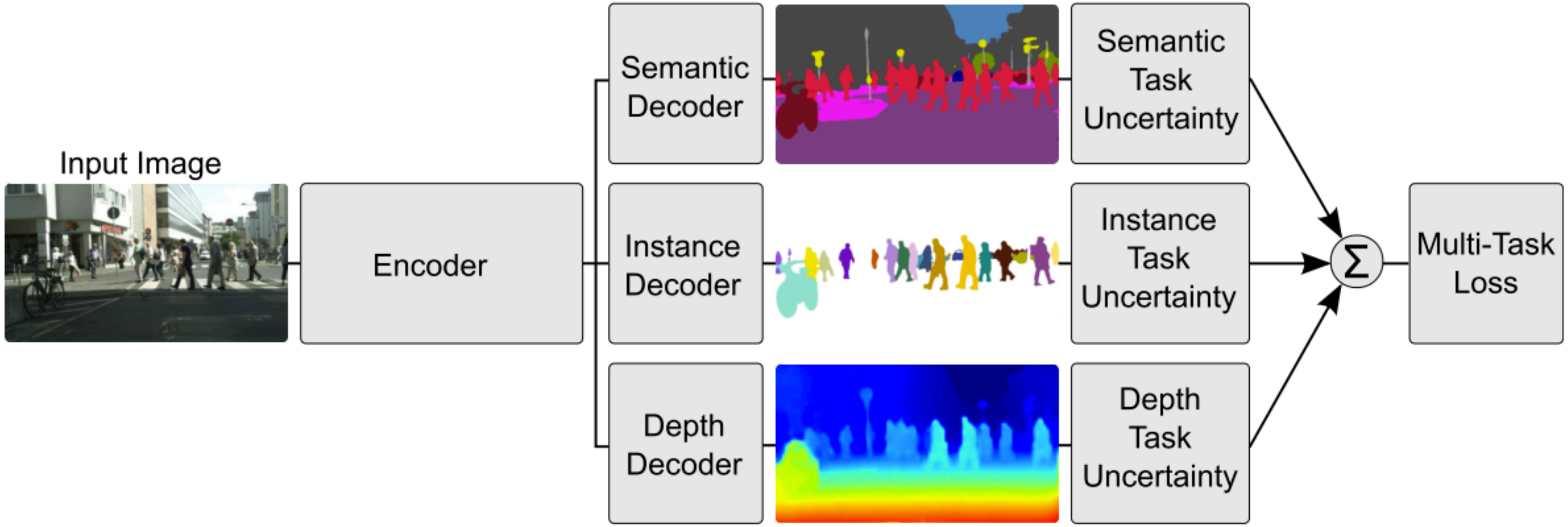
- Tasks & Domains
- What is Transfer Learning?
- Transfer Learning Methods
    - Instance Transfer
    - Feature Transfer
    - Parameter Transfer
- ***Multi-Task Learning***

Multi-Task Learning

- Jointly train multiple related tasks
  - Improves performance of all tasks by knowledge transfer
  - Reduce inference cost by sharing network weights
- Multi-task Learning (MTL) is a subset of transfer learning
  - We care about performance of all tasks
  - Tasks are trained jointly

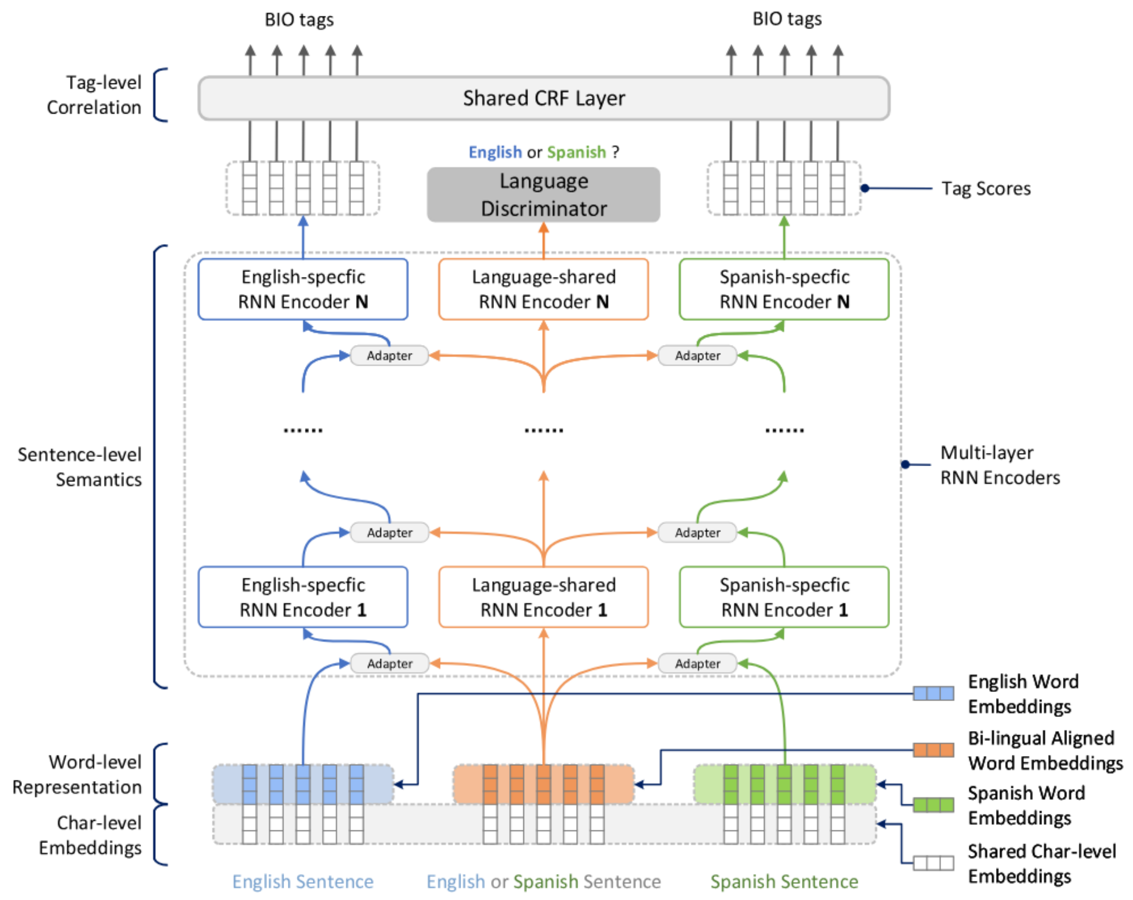


Examples



Kendall, Alex, Yarin Gal, and Roberto Cipolla. "Multi-task learning using uncertainty to weigh losses for scene geometry and semantics." Proceedings of the IEEE conference on computer vision and pattern recognition. 2018.

Examples

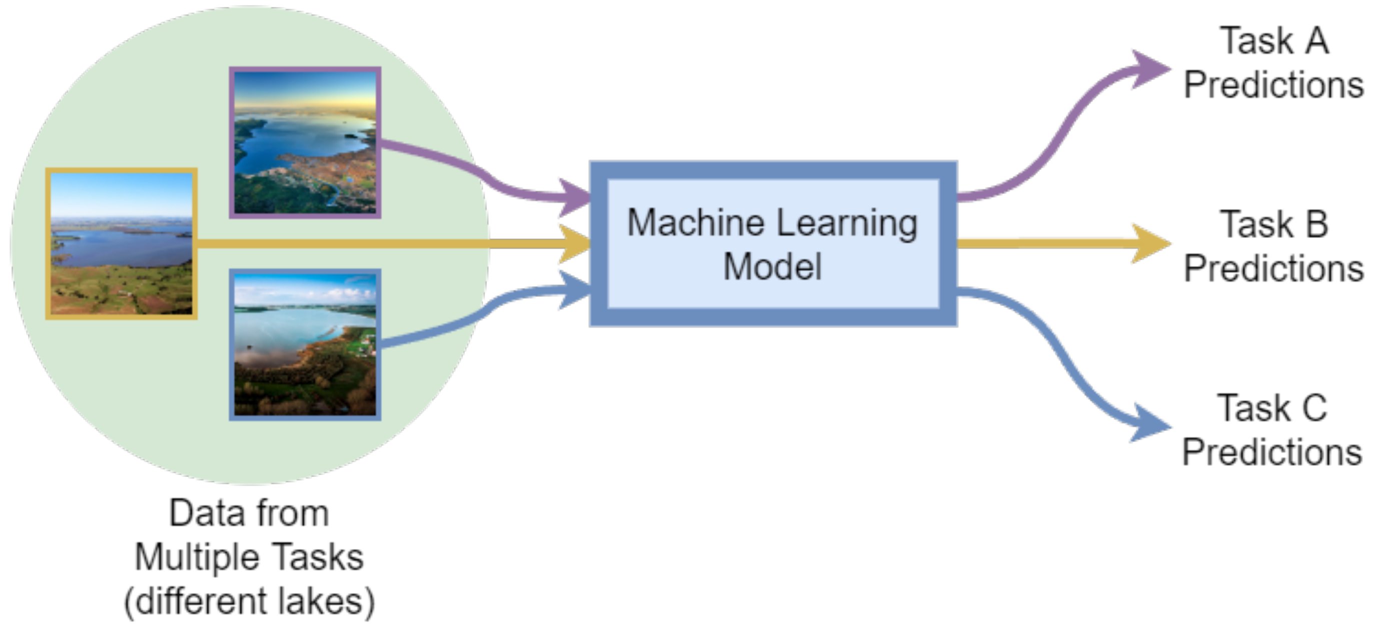

BIO tags
BIO tags
Tag-level Correlation
Shared CRF Layer
English or Spanish ?
Language Discriminator
Tag Scores
English-specfic RNN Encoder N
Language-shared RNN Encoder N
Spanish-specfic RNN Encoder N
Adapter
Sentence-level Semantics
Multi-layer RNN Encoders
English-specfic RNN Encoder 1
Language-shared RNN Encoder 1
Spanish-specfic RNN Encoder 1
English Word Embeddings
Bi-lingual Aligned Word Embeddings
Spanish Word Embeddings
Shared Char-level Embeddings
Word-level Representation
Char-level Embeddings
English Sentence
English or Spanish Sentence
Spanish Sentence

He, K., Xu, W., & Yan, Y. (2020). Multi-level cross-lingual transfer learning with language shared and specific knowledge for spoken language understanding. *IEEE Access*, *8*, 29407-29416.

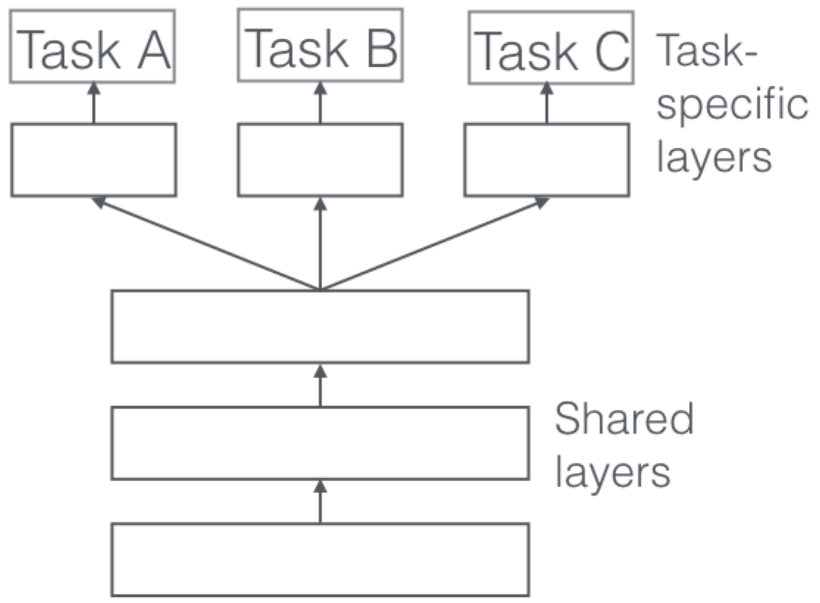
Multi-Task Learning

THE UNIVERSITY OF AUCKLAND
Te Whare Wānanga o Tāmaki Makaurau
NEW ZEALAND

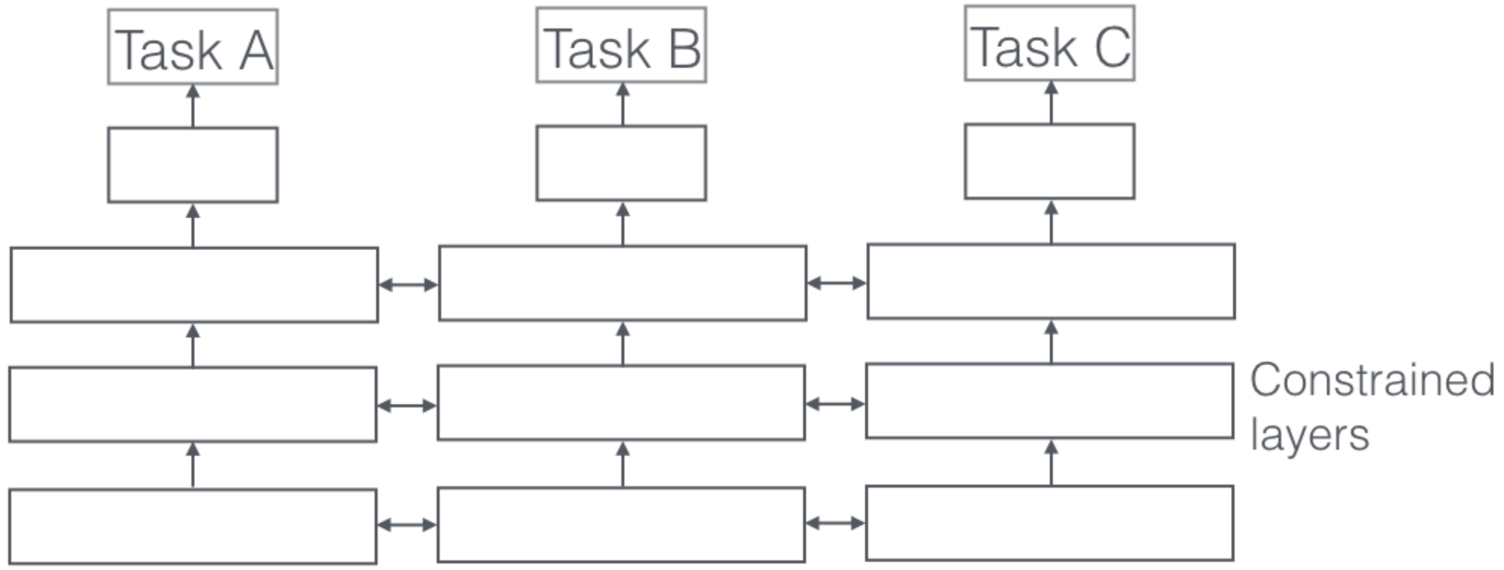
Examples



Techniques



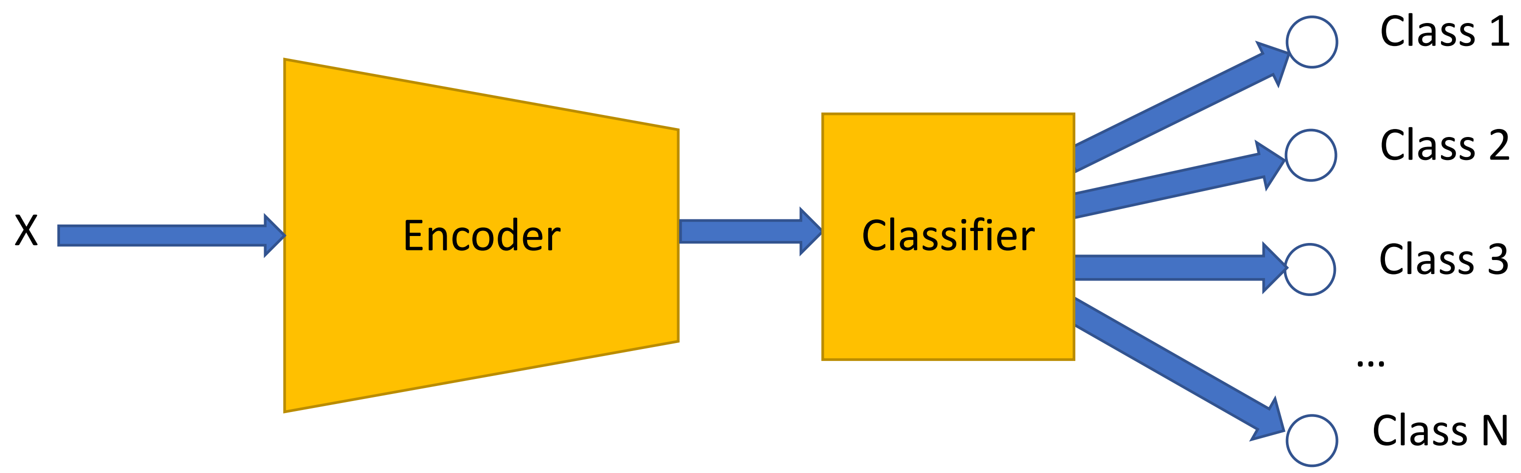
Hard parameter sharing



Soft parameter sharing

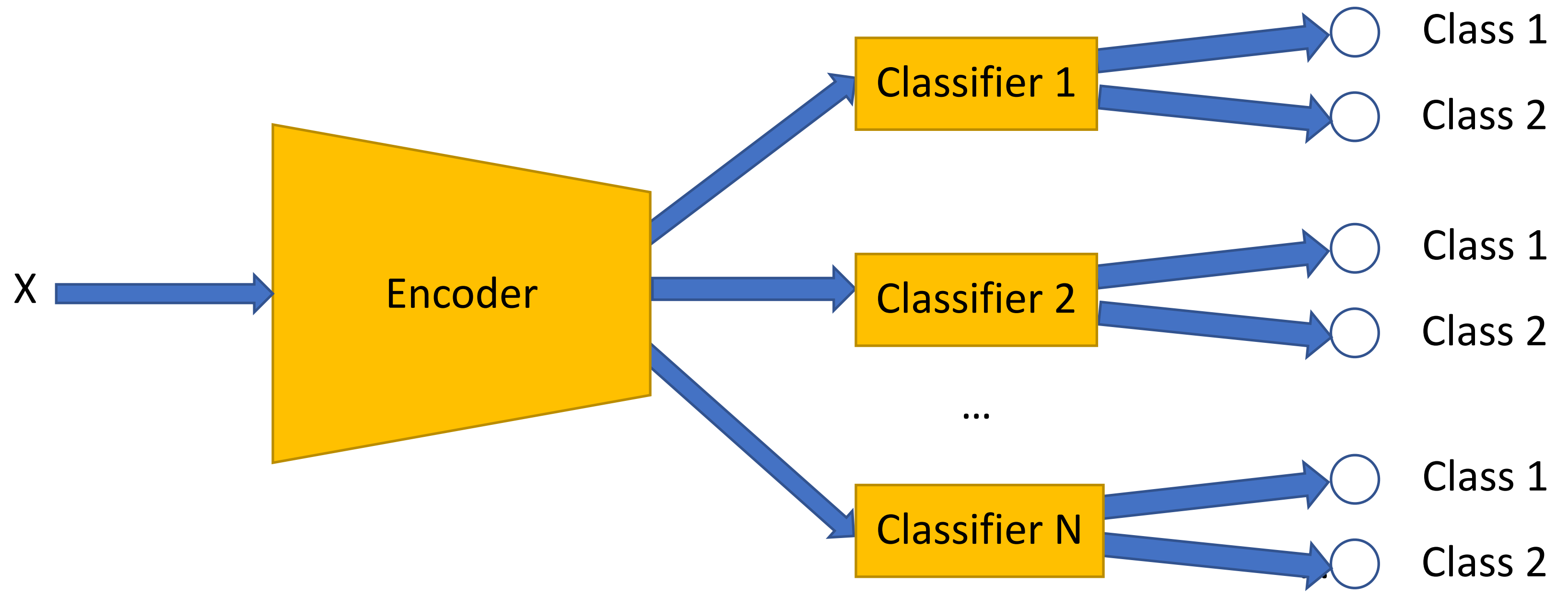
Hard-Parameter Sharing

Single-task learning architecture:



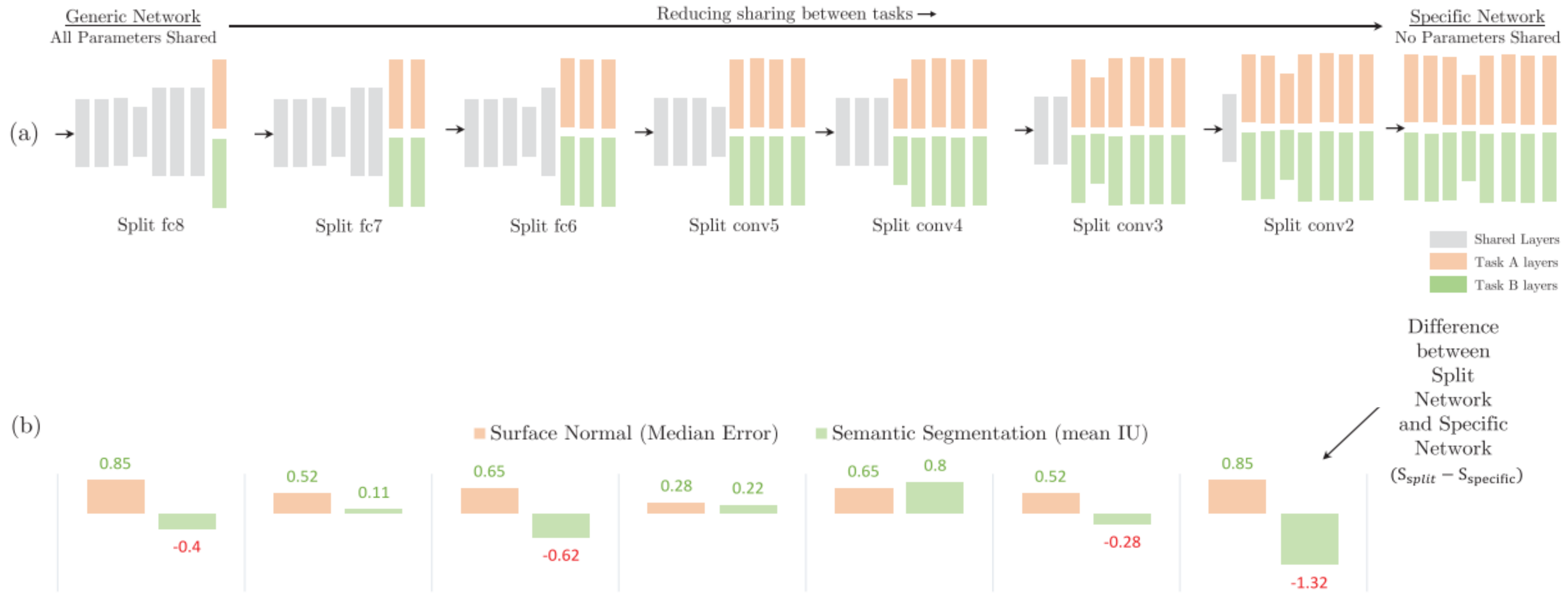
Hard-Parameter Sharing

Multi-task learning architecture: (hard-sharing)



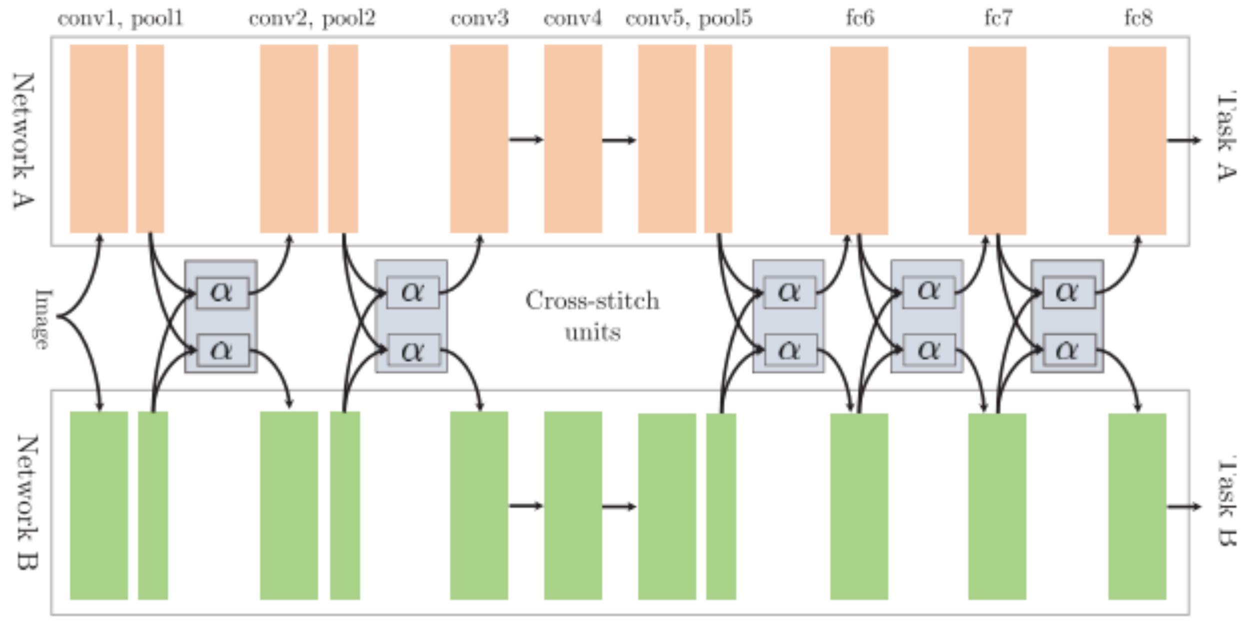
Hard-Parameter Sharing

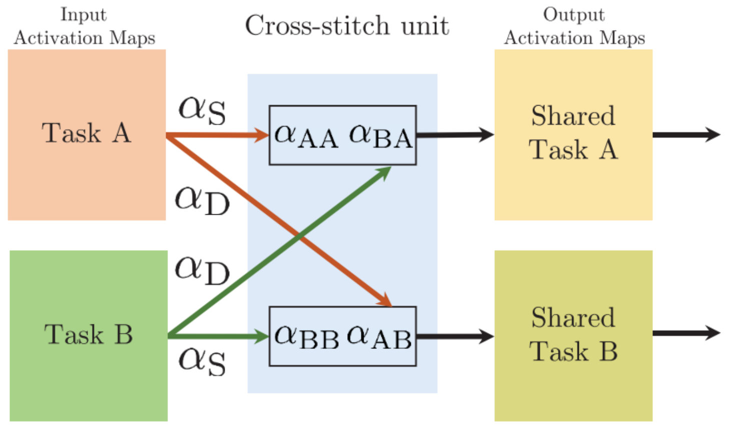
What layers should we share?



Cross-Stitch Networks

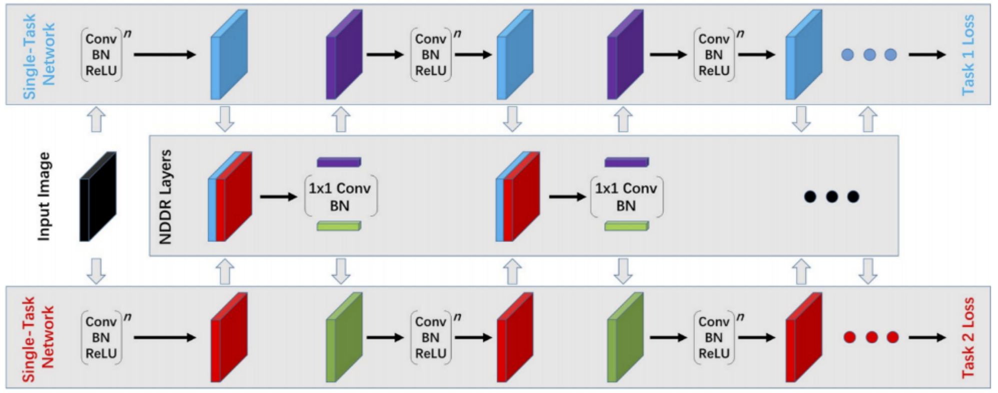
What layers should we share?
Let the model decide!

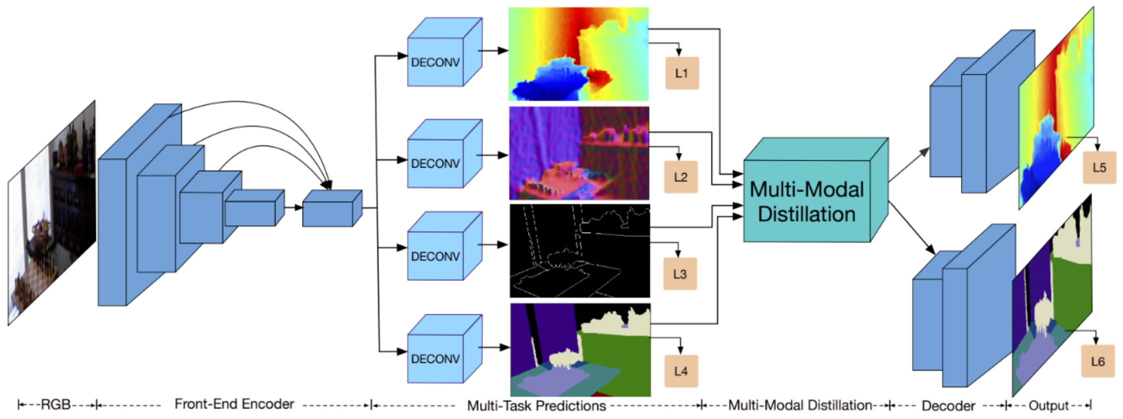


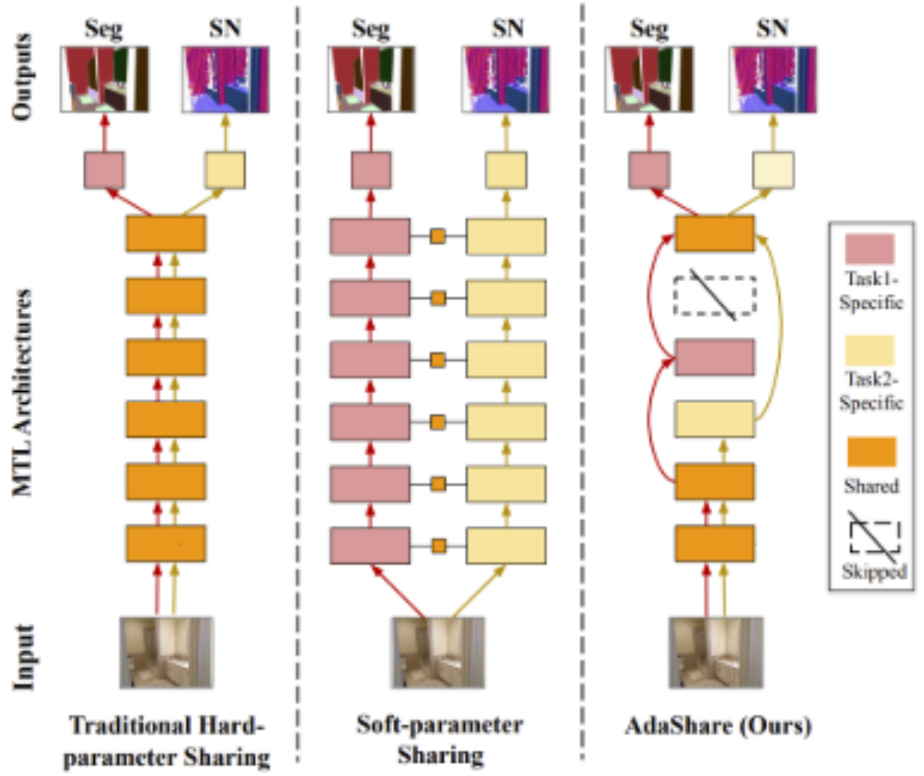


"Cross-stitch" layers compute linear combinations of intermediate representations

Other MTL architectures







Crawshaw, M. (2020). Multi-task learning with deep neural networks: A survey. *arXiv preprint arXiv:2009.09796*.

Multi-Task Learning Optimisation

Q: How do we train a model with multiple tasks?

A: We have multiple loss functions!

Q: How do we weigh these loss functions?

A: [1000 different proposed techniques]

Task-specific loss weight coefficient

$$\min_{\boldsymbol{\theta}} \; L_i\left(f\left(x_i; \theta_{sh}, \theta_i\right), y_i\right),$$

STL optimisation

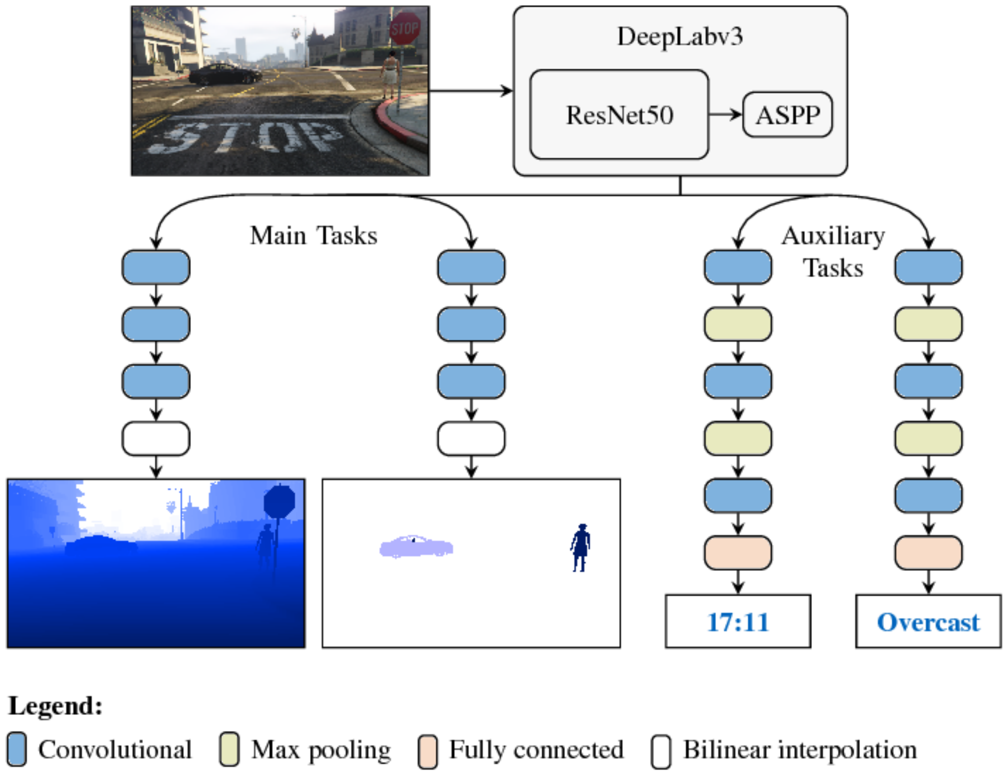
$$\min_{\boldsymbol{\theta}} \sum_{i=1}^{K} \lambda_i \cdot L_i\left(f\left(x_i; \theta_{sh}, \theta_i\right), y_i\right),$$

MTL optimisation

Liu, S., James, S., Davison, A., & Johns, E. (2022). Auto-Lambda: Disentangling Dynamic Task Relationships. *Transactions on Machine Learning Research*.

Auxiliary Learning

- If learning multiple related tasks together improves learning, why not learn extra (auxiliary) tasks along with the task(s) we care about?



MTL in the context of LLMs

- Traditionally, NLP models learnt multiple tasks simultaneously
  - *e.g. QA + text classification*
- Next-token prediction (used to train LLMs) is a single task but enables these models to excel in many tasks
  - This reduces the importance of MTL for NLP

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., ... & Amodei, D. (2020). Language models are few-shot learners. Advances in neural information processing systems, 33, 1877-1901.