

Robot Learning

Meta & Multi-task Learning

Project milestone report

- 5% of the overall class grade (=12.5% of the project grade)
- 2 pages
- You don't have to re-introduce the project. You can assume we already know it, unless...
 - You partially and fully changed your project after the proposal report.
 - Our feedback to your proposal included some clarification questions.

Project milestone report

- Describe the progress since the proposal report.
 - Did you complete the mathematical formulation / solution? Share your work.
 - Did you implement something? Share any results you may have.
 - Did you read many papers? Share what you learned and how they are related to your project.
- Describe what else remains to be done.

Today...

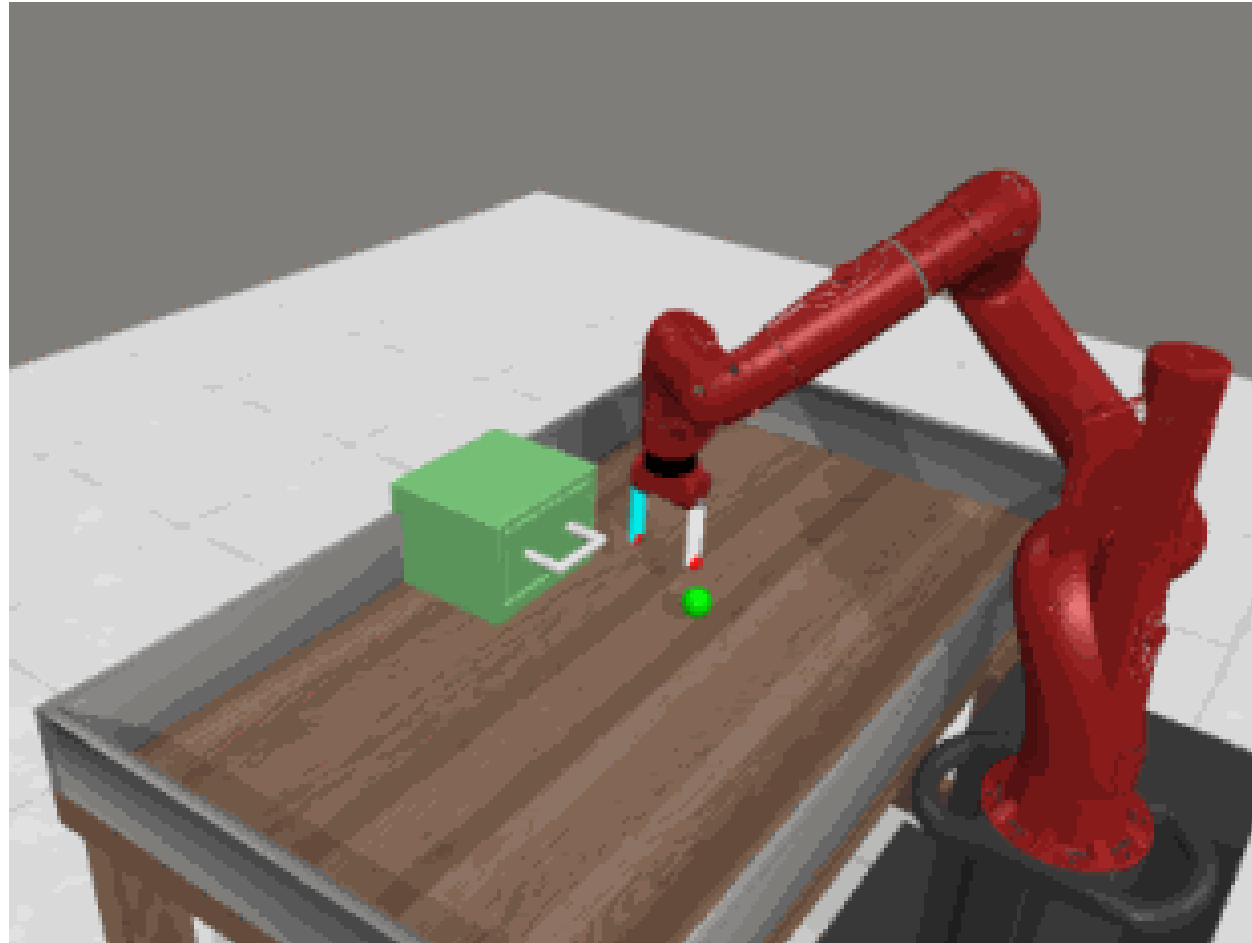
- Multi-task Learning
- Transfer Learning
- Meta Learning

What is a task?

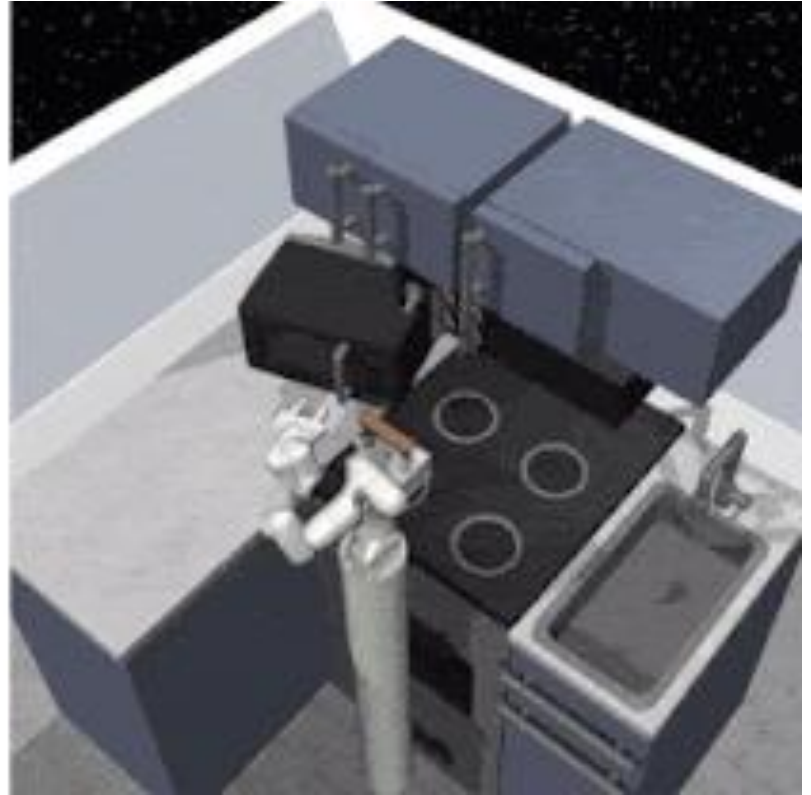
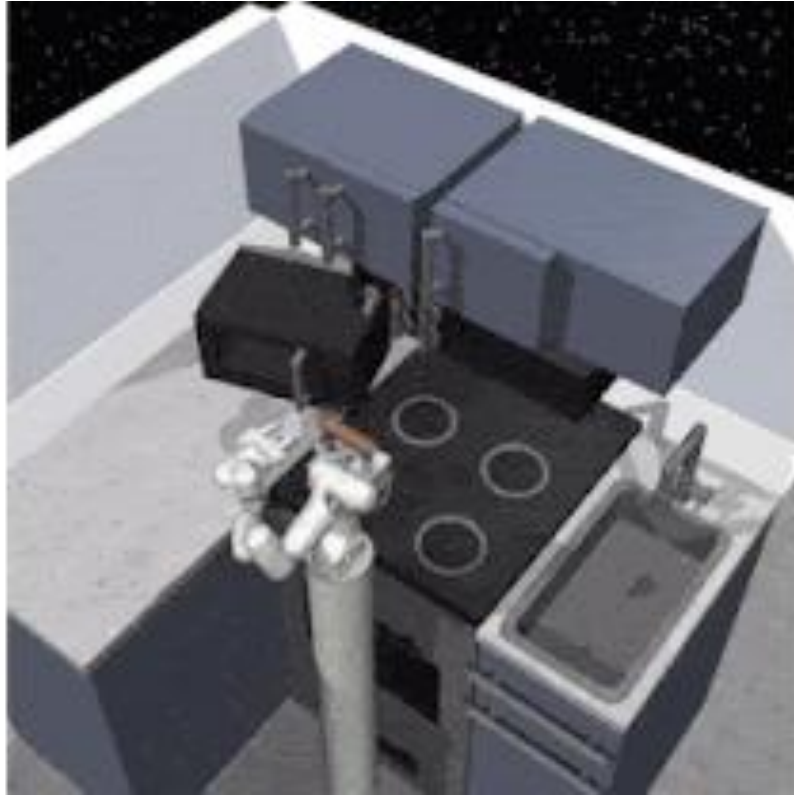
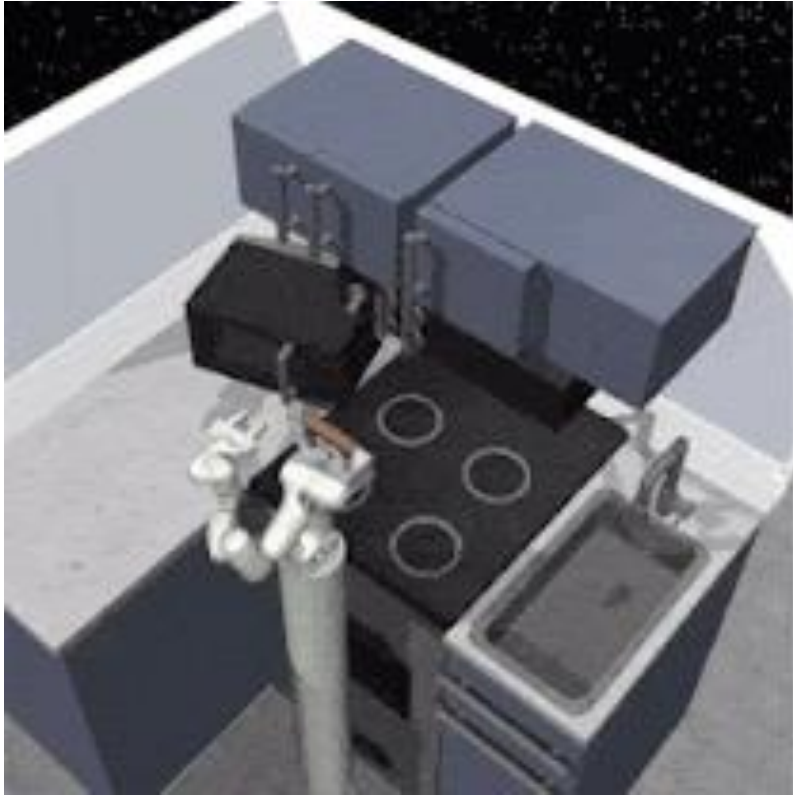
$$\begin{aligned} & \underset{\pi}{\text{maximize}} && \mathbb{E}_{\mathbf{w}} \left[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t) \right] \\ & \text{subject to} && s_{t+1} = f(s_t, a_t, \mathbf{w}_t) \\ & && a_t = \pi(s_t) \end{aligned}$$

More generally in machine learning, a dataset-loss function pair defines a task.

What's wrong with single-task learning?



What's wrong with single-task learning?



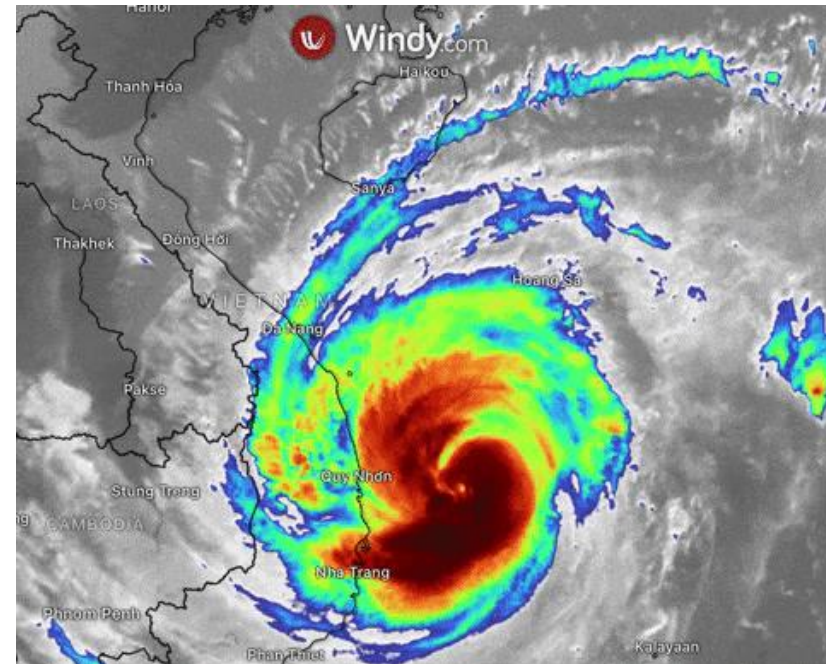
Still... Why multi-task learning?

We could just learn each task independently!

- What if we have little data for some tasks?
- What if we have little time to learn some tasks?

Still... Why multi-task learning?

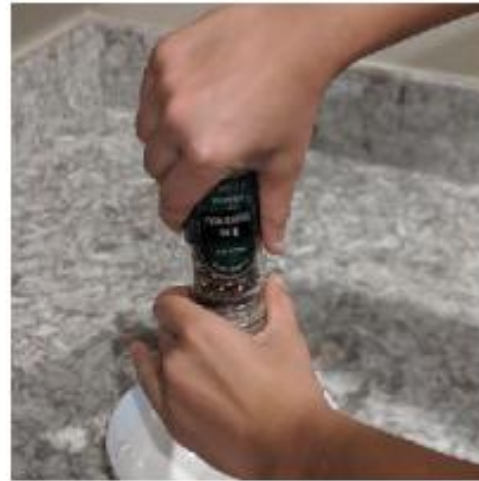
You should learn each task independently if there is no shared structure between the tasks.



Left: Playing atari with deep reinforcement learning
Mnih et al., NeurIPS Deep Learning Workshop 2013
Right: Windy.com Community

Still... Why multi-task learning?

In real life, many tasks share structure!



Today...

- Multi-task Learning
- Transfer Learning
- Meta Learning

Today...

- Multi-task Learning: Learn multiple tasks together
- Transfer Learning
- Meta Learning

Today...

- Multi-task Learning: Learn multiple tasks together
- Transfer Learning: Learn multiple tasks and transfer your knowledge to a new one
- Meta Learning

Today...

- Multi-task Learning: Learn multiple tasks together
- Transfer Learning: Learn multiple tasks and transfer your knowledge to a new one
- Meta Learning: Learn multiple tasks such that adapting to a new task will be easy

Today...

- Multi-task Learning: Learn multiple tasks together
- Transfer Learning: Learn multiple tasks and transfer your knowledge to a new one
- Meta Learning: Learn multiple tasks such that adapting to a new task will be easy

Multi-task learning

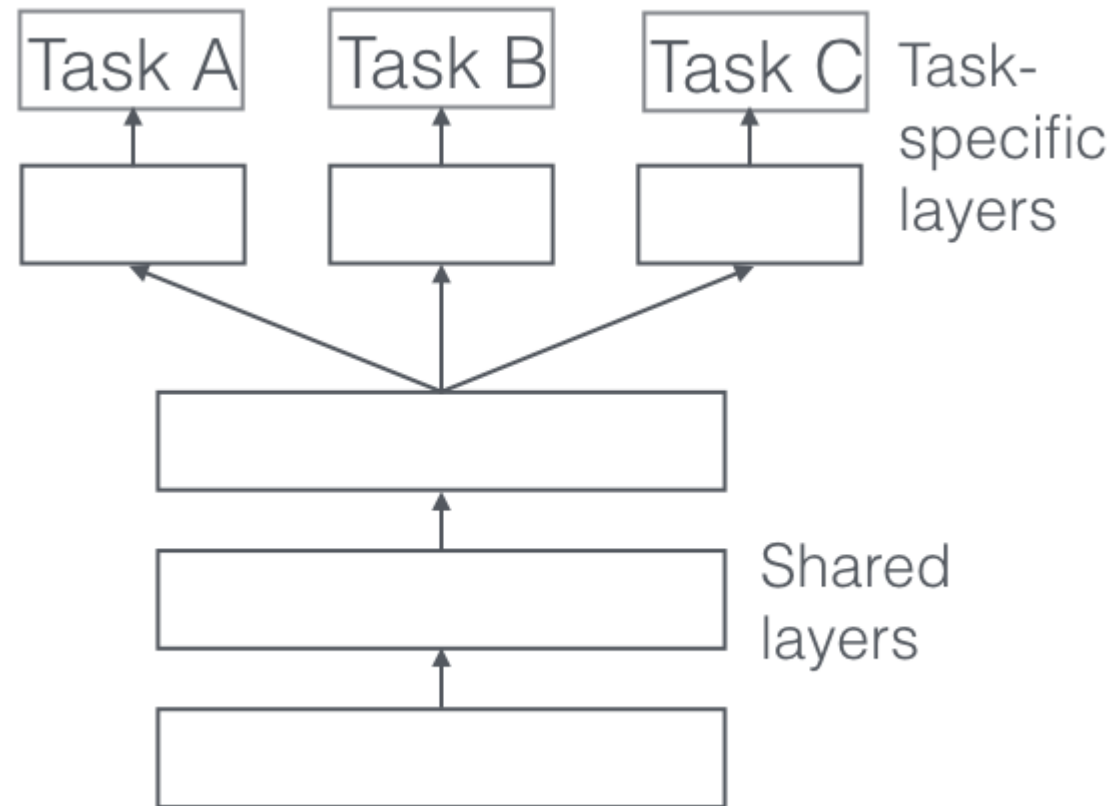
Common solution:

1. Sample tasks from the **task distribution** $P(\tau)$
2. Compute their losses
3. Sum the losses
4. Backpropagate
5. Go back to step 1

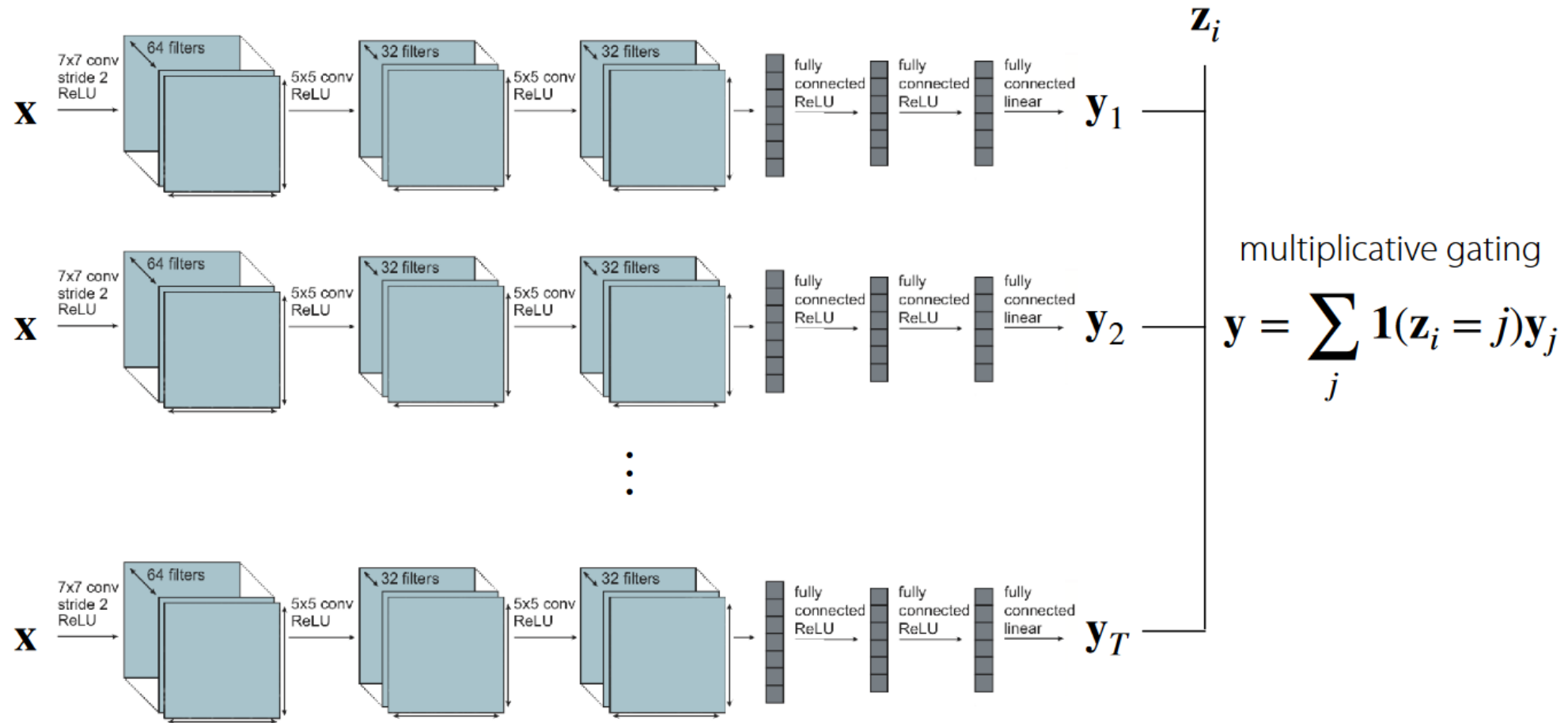
How will
the model
know which
task to do?

What if the
sum of
losses is not
a good loss?

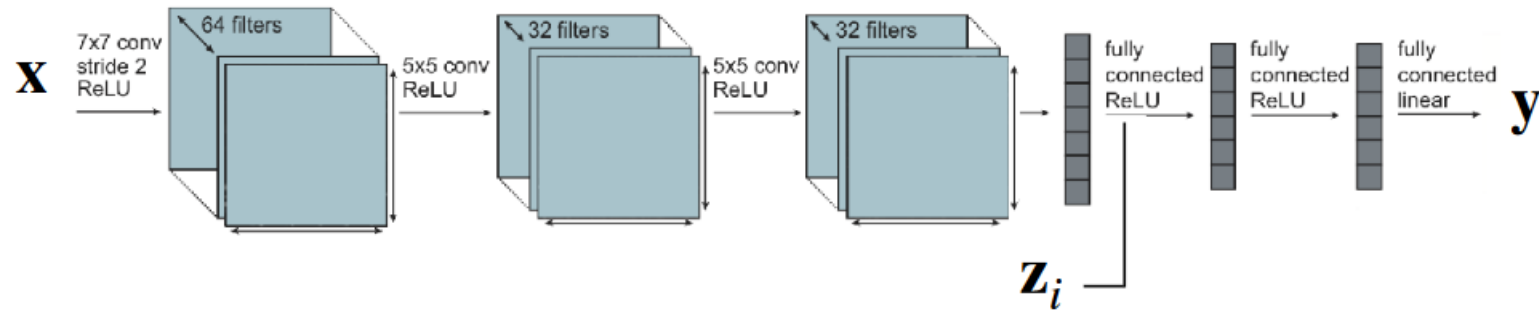
Parameter sharing



Extreme case: no parameter sharing

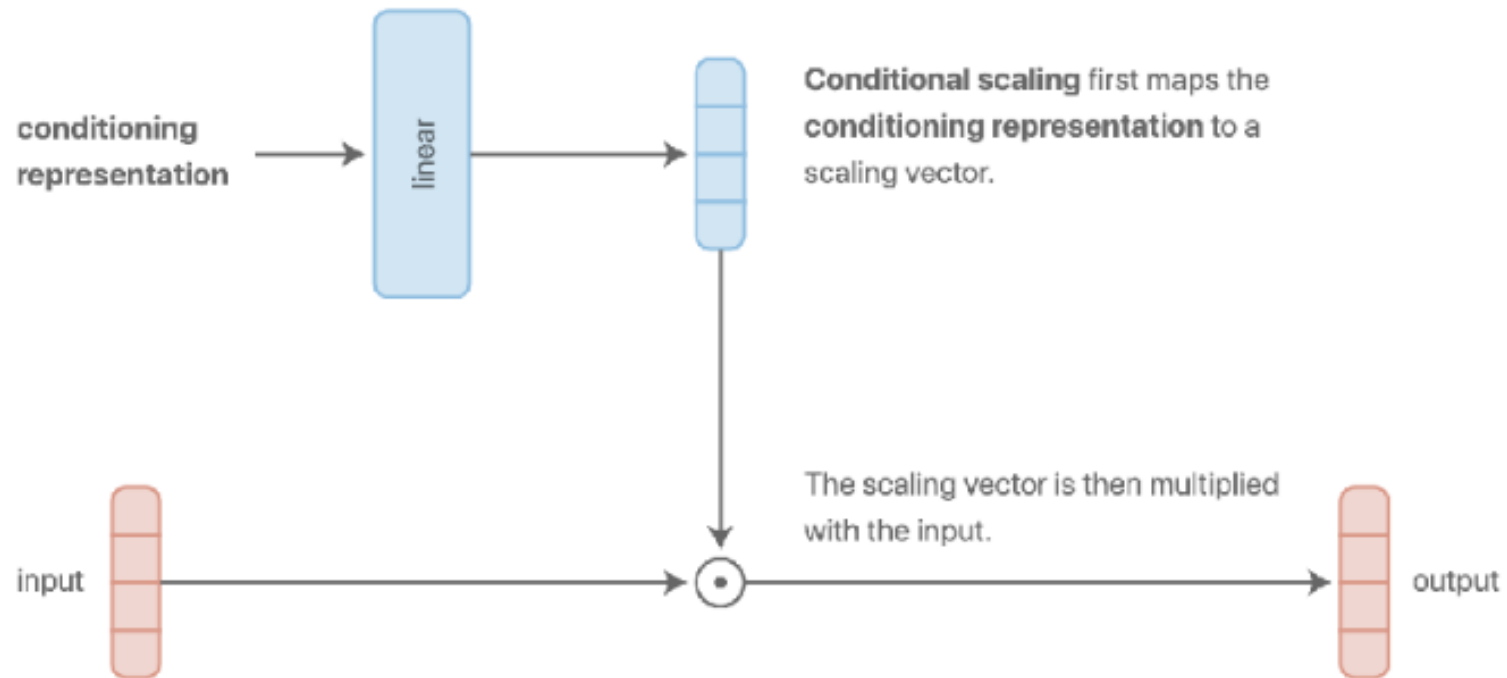


Extreme case: full parameter sharing

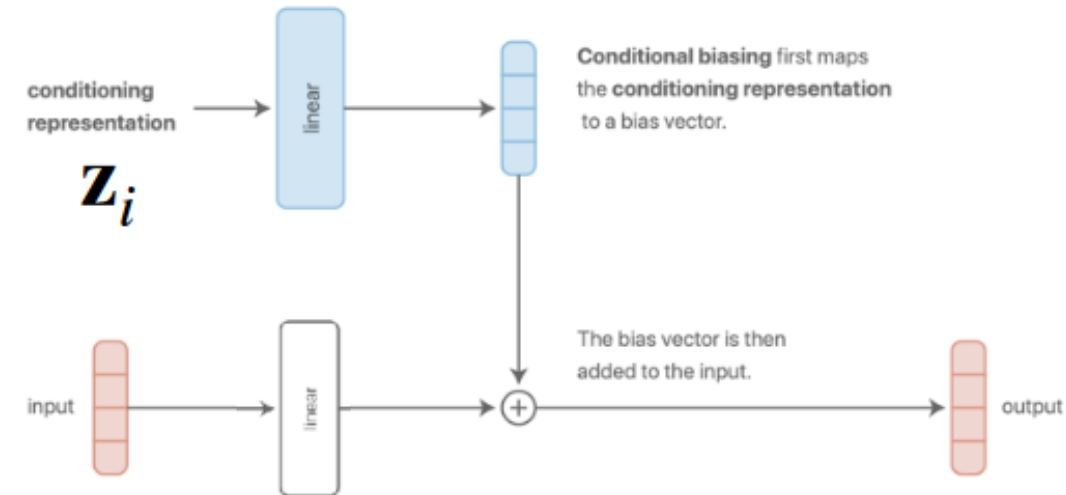
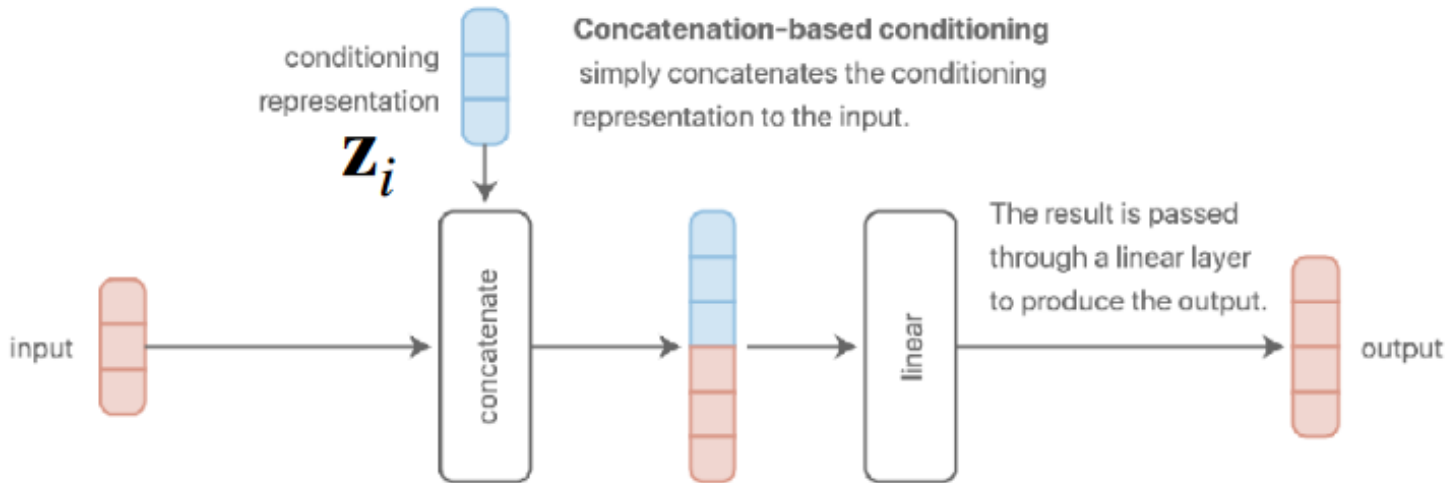


Concatenate \mathbf{z}_i with input and/or activations

Multiplicative coding

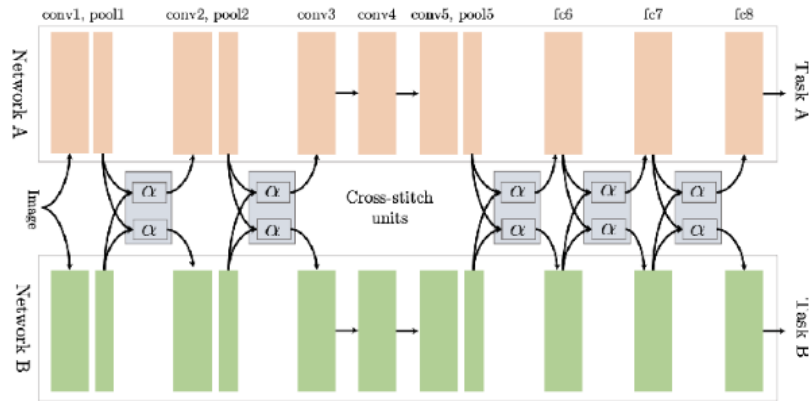


Concatenation-based coding

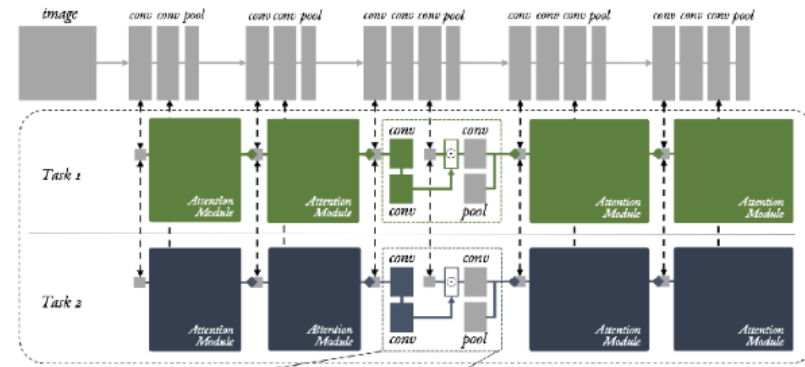


These are the same!

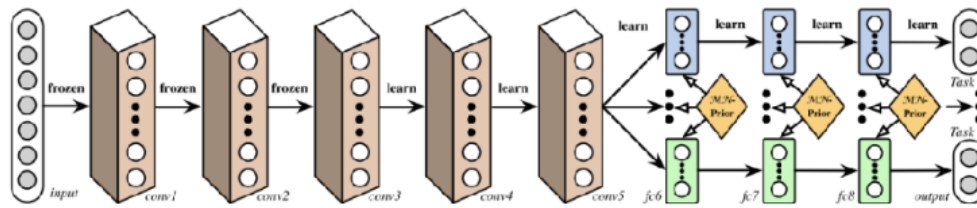
There is no right way



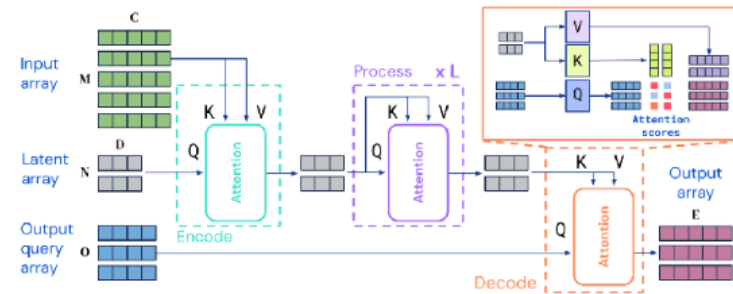
Cross-Stitch Networks. Misra, Shrivastava, Gupta, Hebert '16



Multi-Task Attention Network. Liu, Johns, Davison '18



Deep Relation Networks. Long, Wang '15



Perceiver IO. Jaegle et al. '21

Multi-task learning

Common solution:

1. Sample tasks from the task distribution $P(\tau)$
2. Compute their losses
3. Sum the losses
4. Backpropagate
5. Go back to step 1

How will
the model
know which
task to do?



What if the
sum of
losses is not
a good loss?

Multi-task learning

Common solution:

1. Sample tasks from the task distribution $P(\tau)$
2. Compute their losses
3. Sum the losses *w/ some weights*
4. Backpropagate
5. Go back to step 1

Popular heuristic: try to make gradients have similar magnitude

How will the model know which task to do?

What if the sum of losses is not a good loss?



Multi-task learning

Common solution:

1. Sample tasks from the task distribution $P(\tau)$
2. Compute their losses
3. **Take the maximum of** the losses
4. Backpropagate
5. Go back to step 1

How will
the model
know which
task to do?

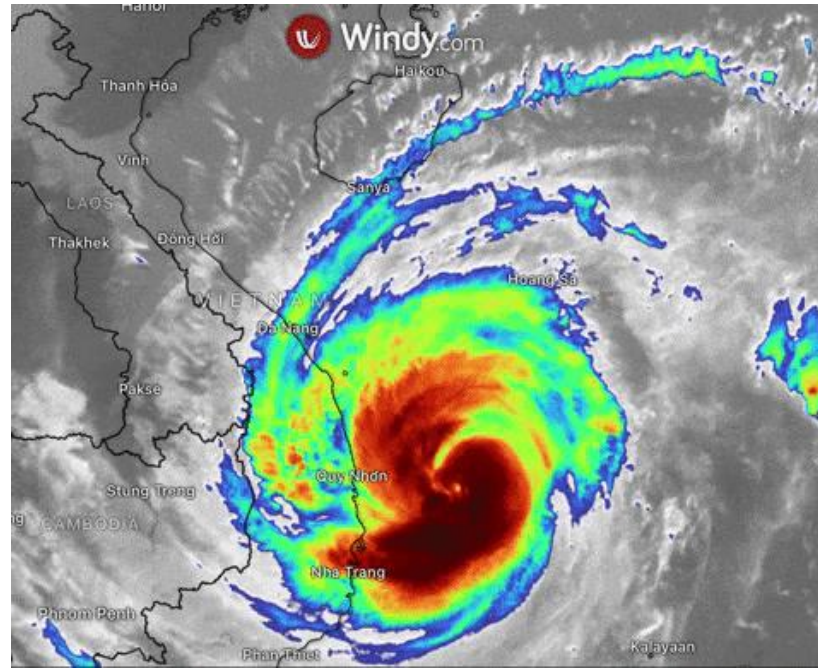


What if the
sum of
losses is not
a good loss?



Common problems

- Negative transfer



You should share less between the tasks.

How?

- Fewer parameters
- Soft-sharing

Soft-sharing

Do not constrain the model to have the same parameters for different tasks.

Instead, penalize the model based on how different their parameters are.

Common problems

- Overfitting

Perhaps, you have little data for some of the tasks.

You should share more between the tasks.

Can we share based on task similarity!

Yes!

But what is task similarity?

Today...

- Multi-task Learning: Learn multiple tasks together
- Transfer Learning: Learn multiple tasks and transfer your knowledge to a new one
- Meta Learning: Learn multiple tasks such that adapting to a new task will be easy

Transfer learning

Training: Have access to tasks $\tau_1, \tau_2, \dots, \tau_n$, but not τ_{n+1} .

Transfer: Have access to task τ_{n+1} , but not $\tau_1, \tau_2, \dots, \tau_n$.

Transfer learning

Common solution:

Training:

1. Run your favorite (multi-task) learning algorithm on $\tau_1, \tau_2, \dots, \tau_n$

Transfer:

2. Fine-tune the model on τ_{n+1}

This is the idea behind using ImageNet features or BERT embeddings!

Transfer learning

Common solution:

Training:

1. Run your favorite (multi-task) learning algorithm on $\tau_1, \tau_2, \dots, \tau_n$

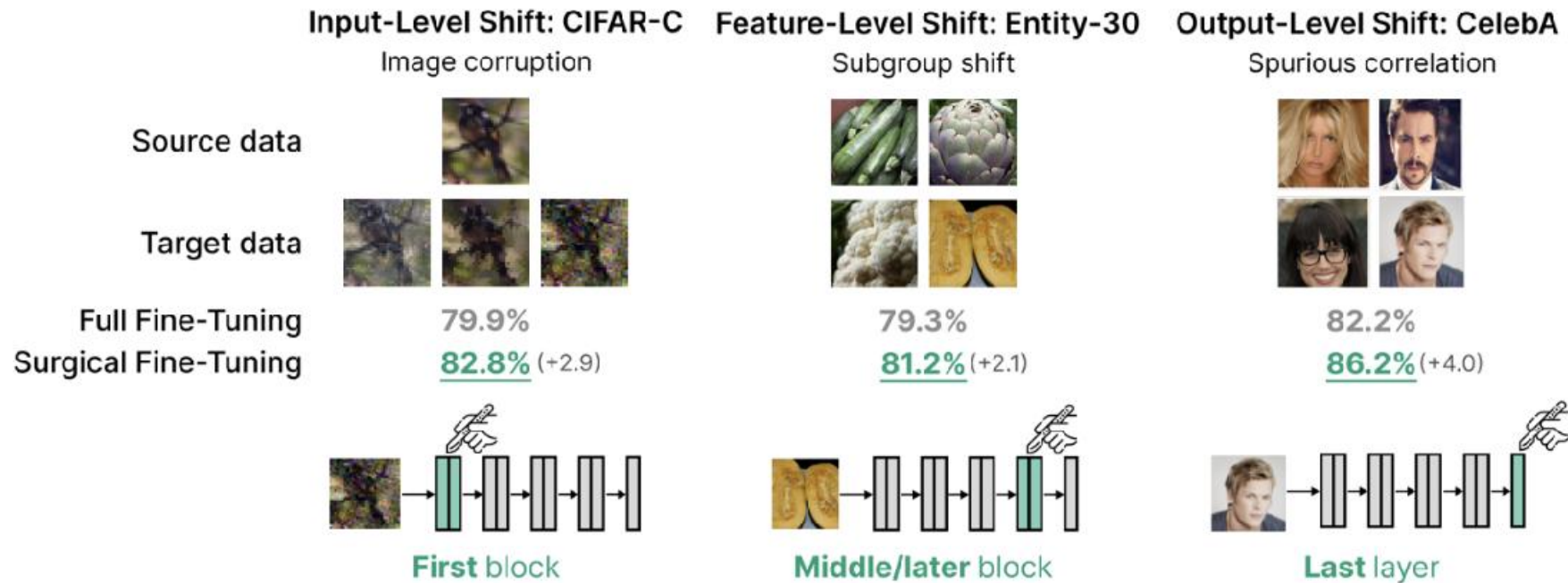
Transfer:

2. Fine-tune the model on τ_{n+1}

Fine-tune
what?

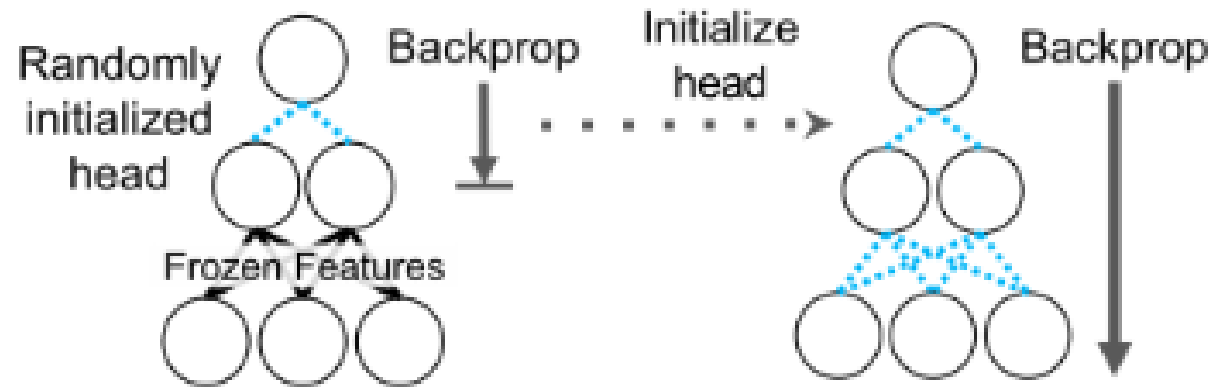
Fine-tune what?

It depends.



Fine-tune what?

A good default:



Transfer learning

What if our dataset on the target set is so small that even transfer learning does not help?

Today...

- Multi-task Learning: Learn multiple tasks together
- Transfer Learning: Learn multiple tasks and transfer your knowledge to a new one
- Meta Learning: Learn multiple tasks such that adapting to a new task will be easy

Meta-learning



Given 1 example of 5 classes:

Classify new examples



Meta-learning

Training: Have access to tasks $\tau_1, \tau_2, \dots, \tau_n$, but not τ_{n+1} .

Transfer: Have access to task τ_{n+1} , but not $\tau_1, \tau_2, \dots, \tau_n$.

Assumption:

τ_{n+1} comes from the same task distribution as $\tau_1, \tau_2, \dots, \tau_n$.

Black-box adaptation

- Design a giant neural network that takes the datasets as the input and outputs the parameters of a smaller network.

Yes, I really said this.

But sometimes we can get away with lower dimensional vectors.

- The smaller network performs the task τ_{n+1} .

Optimization-based adaptation

Learn a model such that when we take one (or some) gradient step in task τ_{n+1} , it will perform good.

$$\underset{\theta}{\text{minimize}} \sum_{i=1}^n L(\theta - \alpha \nabla_{\theta} L(\theta, \tau_i^{tr}), \tau_i^{ts})$$

Optimization-based adaptation

1. Sample task τ_i
 2. Compute $\phi \leftarrow \theta - \nabla_{\theta} L(\theta, \tau_i^{tr})$
 3. Update θ using $\nabla_{\theta} L(\phi, \tau_i^{ts})$
- Note we will need the second gradient!

Next time...

Week 11	<i>Guest Lecture</i>	Prof. Heather Culbertson	TBD
Fri, Nov 8	<i>Guest Lecture</i>	Dr. Aaquib Tabrez	Multimodal Explanation-based Reward Coaching and Decision Support to Improve Human-Robot Teaming