Machine Learning Systems Design

Modeling Pipeline

Lecture 13: Hyperparameter tuning and AutoML

CE 40959 Spring 2023
Ali Zarezade
SharifMLSD.github.io

Agenda

1. Hyperparameter Tuning
2. AutoML

1. Hyperparameter Tuning

Parameter vs Hyperparameter

How you distinguish parameters from hyperparameters?

Parameter vs Hyperparameter

How you name a parameter as a hyperparameter?

- A parameter is optimized by the algorithm, while a hyperparameter is tuned by the engineer.😊
- Every step in your entire machine learning pipeline can have its own hyperparameters.

Parameter vs Hyperparameter

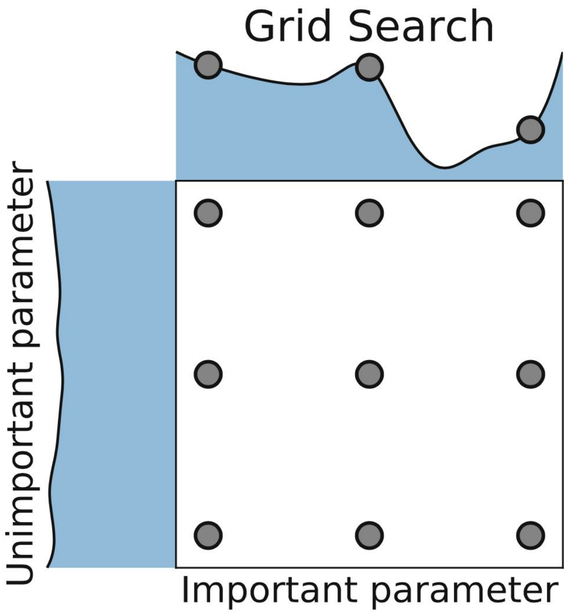
For example, in data **pre-processing**, the hyperparameters could specify whether to use data-augmentation or using which technique to fill missing values. In **feature engineering**, a hyperparameter could define which feature selection technique to apply. In **modeling**, when making predictions with a model that returns a score, a hyperparameter could specify the decision threshold for each class.

How to search for hyperparameters?

- Manual search (GSD: graduate student descent!)
- Grid search
- Random search
- Bayesian optimization
- Evolutionary algorithms
- And so on

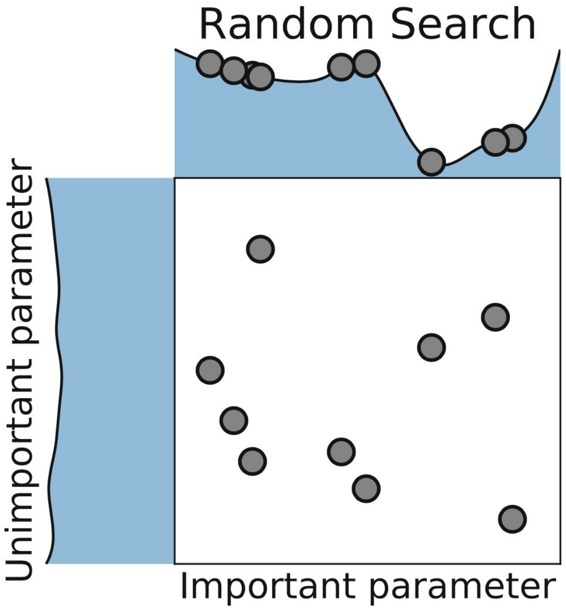
Grid search

It’s used when the number of hyperparameters and their range is not too large.



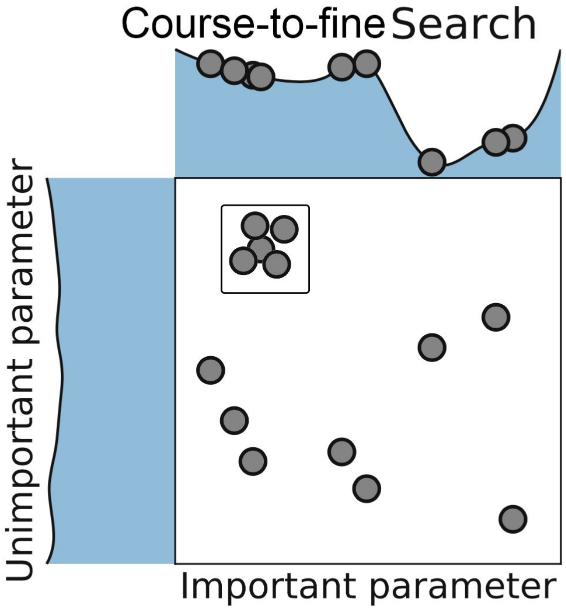
Random search

Here, you provide a statistical distribution for each hyperparameter from which values are randomly sampled. Then set the total number of combinations you want to evaluate.



Coarse-to-fine search

A combination of grid search and random search.



Bayesian optimization

Using a surrogate function and an acquisition function to guide the search for the optimal hyperparameters:

- *Surrogate function*: approximating the true objective function (such as validation accuracy or loss) using the past evaluations of the hyperparameters. A common choice is a Gaussian process (GP), which is a probabilistic model that can model complex and nonlinear functions using a mean function and a covariance function.
- *Acquisition function*: balancing exploration and exploitation to select the next hyperparameters to evaluate. An example is expected improvement (EI), which measures how much improvement one can expect from a new point compared to the best point so far.

Some considerations

Use available **tools** (e.g., Hyperopt) and do not forget experiment tracking.

Hyperparameter tuning discussed above are used when you have a good-sized validation set. When you don’t, a common technique of model evaluation is **cross-validation**.

Some considerations

Sometimes you may see a large gap between dev and train, after hyperparameter tuning.

Why?

Some considerations

You may have **overfit** on dev set by exhaustively searching the search space, or small devset size, or both.

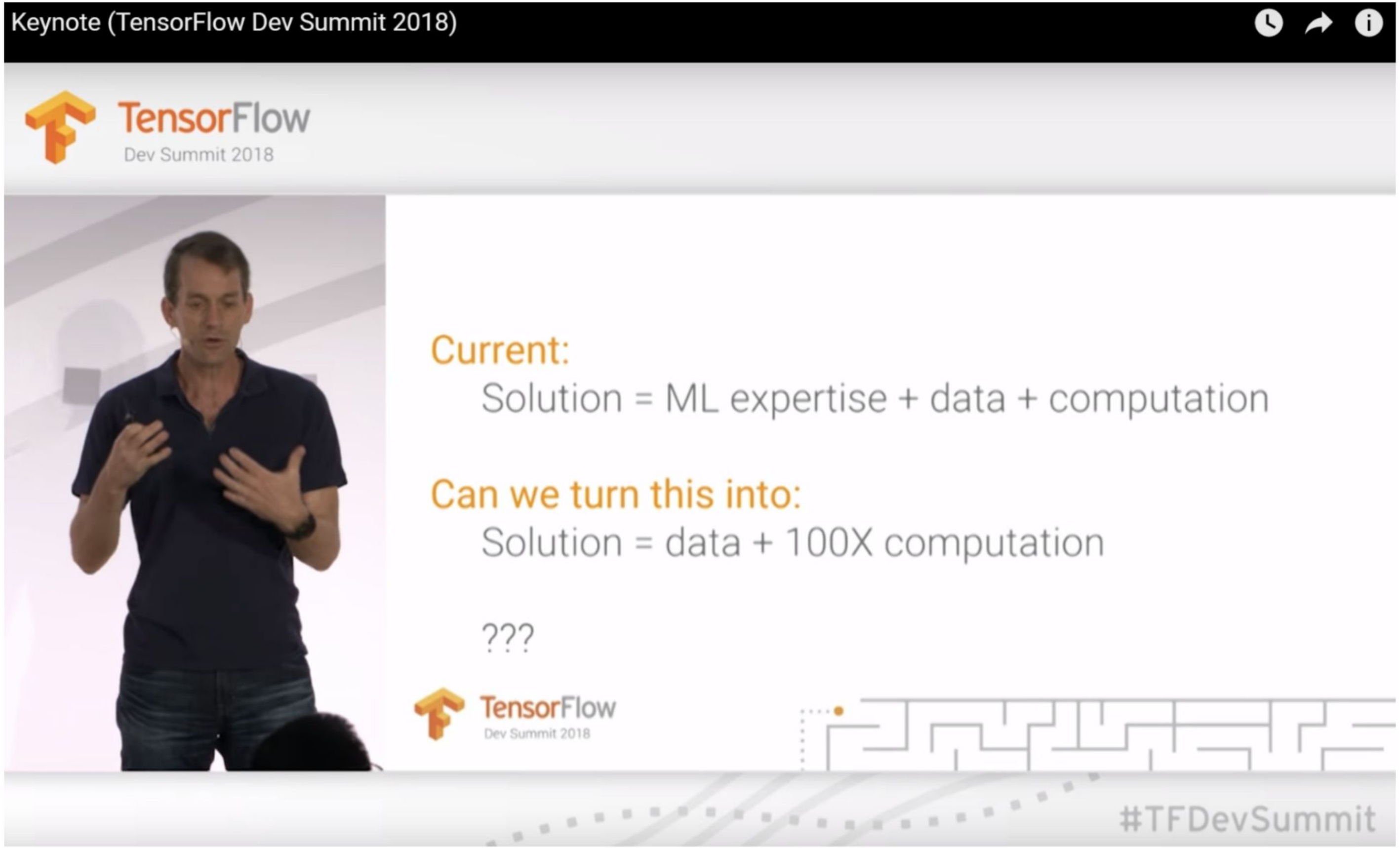
You can resolve this by increasing the dev set or reduce the iteration cycles of your hyperparameter tuning method.

2. AutoML

AutoML

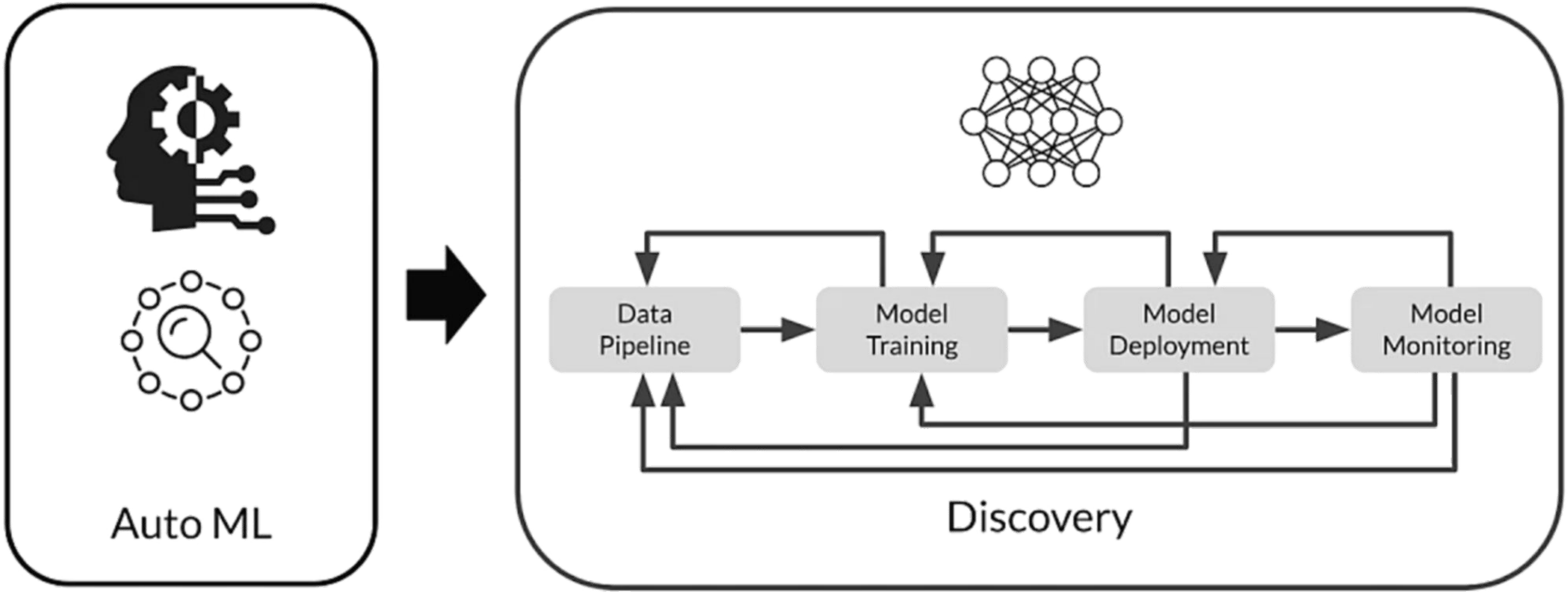
A good ML researcher is someone who will automate themselves out of job :)

AutoML



Keynote (TensorFlow Dev Summit 2018)
TensorFlow
Dev Summit 2018
Current:
Solution = ML expertise + data + computation
Can we turn this into:
Solution = data + 100X computation
???
TensorFlow
Dev Summit 2018
#TFDevSummit

What is AutoML

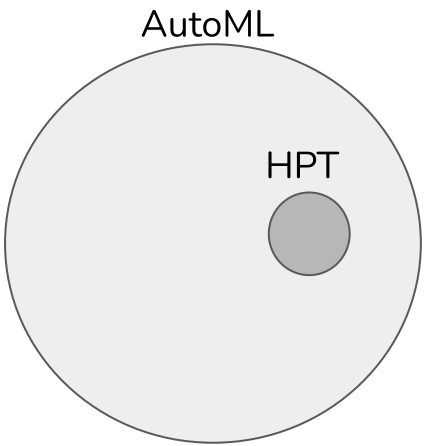


What is AutoML

- It aims to automate the manual tasks involved in machine learning, such as data preprocessing, feature engineering, model selection, hyperparameter tuning, neural architecture search and model deployment.
- It can help users with limited machine learning expertise to train high-quality custom models specific to their business needs.

AutoML vs hyperparameter tuning

- Soft AutoML: Hyperparameter tuning
- Hard AutoML: Architecture search and other pipeline decisions



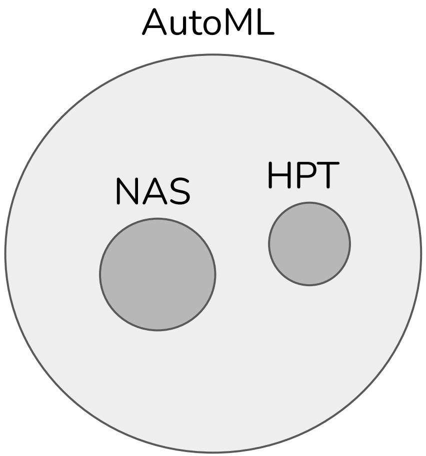
How AutoML works?

AutoML typically involves:

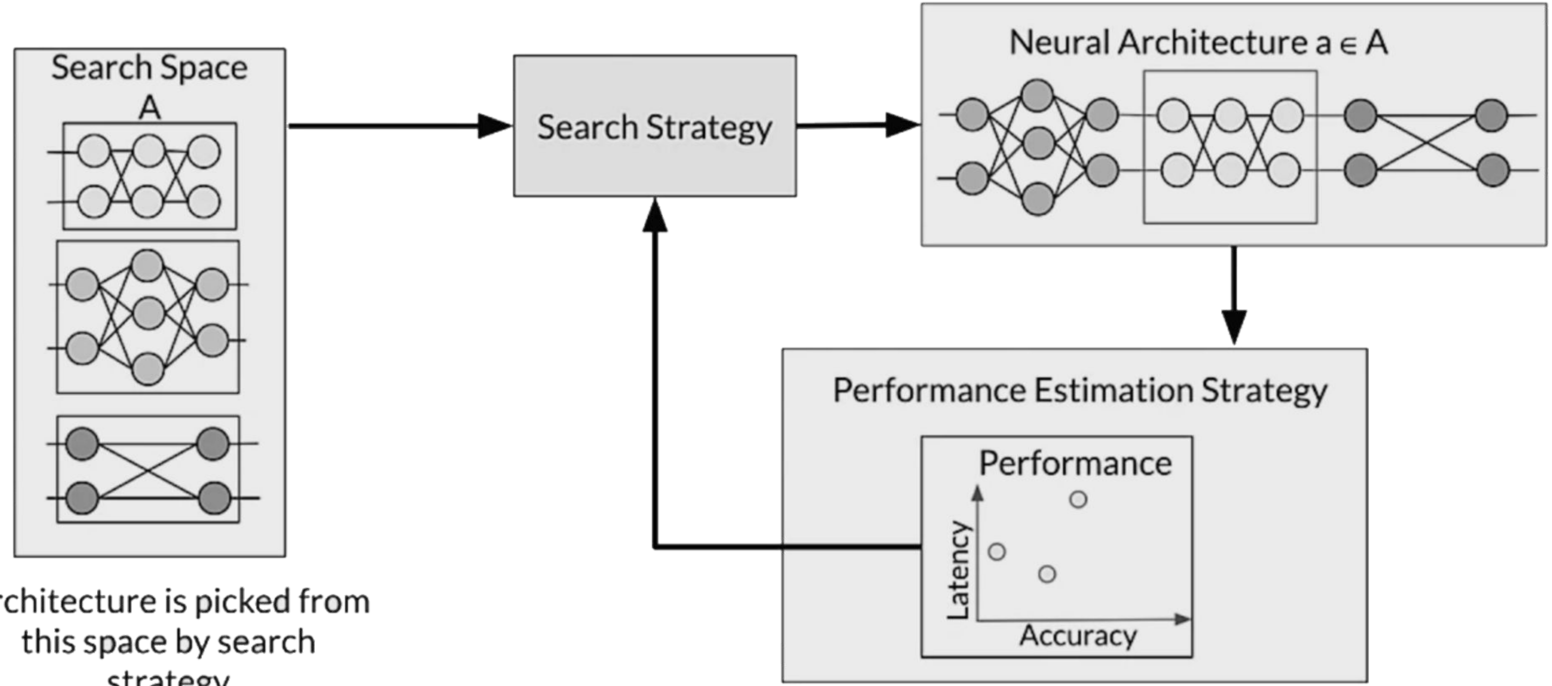
- **Search space**: the possible choices of machine learning components and configurations that can be used to build a machine learning system for a given task.
- **Search strategy**: how to explore the search space efficiently and effectively to find the best machine learning system for the given task (random, RL, Ev.)
- **Performance estimation strategy**: how to evaluate the performance of a candidate architecture without having to train each candidate architecture from scratch until convergence.

NAS: Neural Architecture Search

NAS as a subfield of AutoML is a technique for automating the design of artificial neural networks.



NAS: Neural Architecture Search

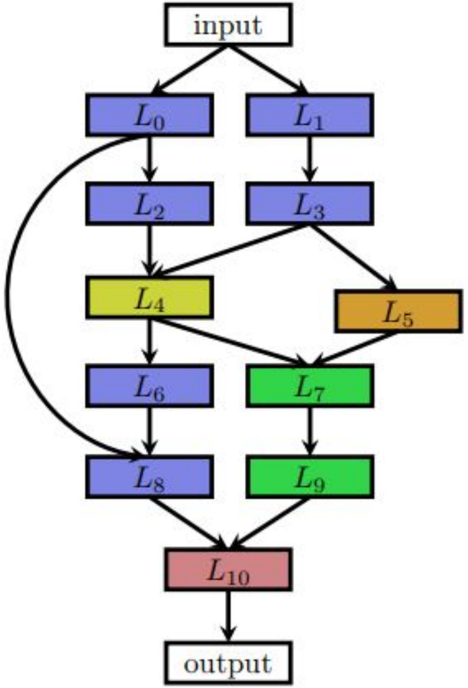


Architecture is picked from this space by search strategy

Search spaces in NAS

Macro architecture search focuses on finding the best way to connect or organize different cells or blocks into a network.

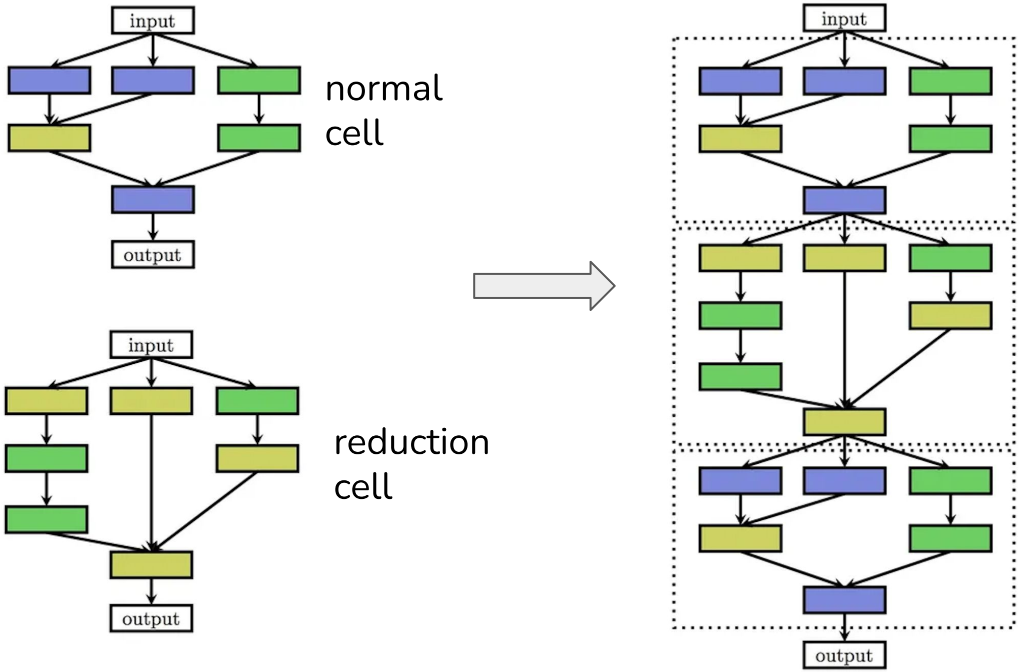
chain structure space



complex search space

Search spaces in NAS

Micro architecture search focuses on finding the best cell or block structure that can be repeated or stacked to form a larger network.



Search strategies in NAS

Some strategies are:

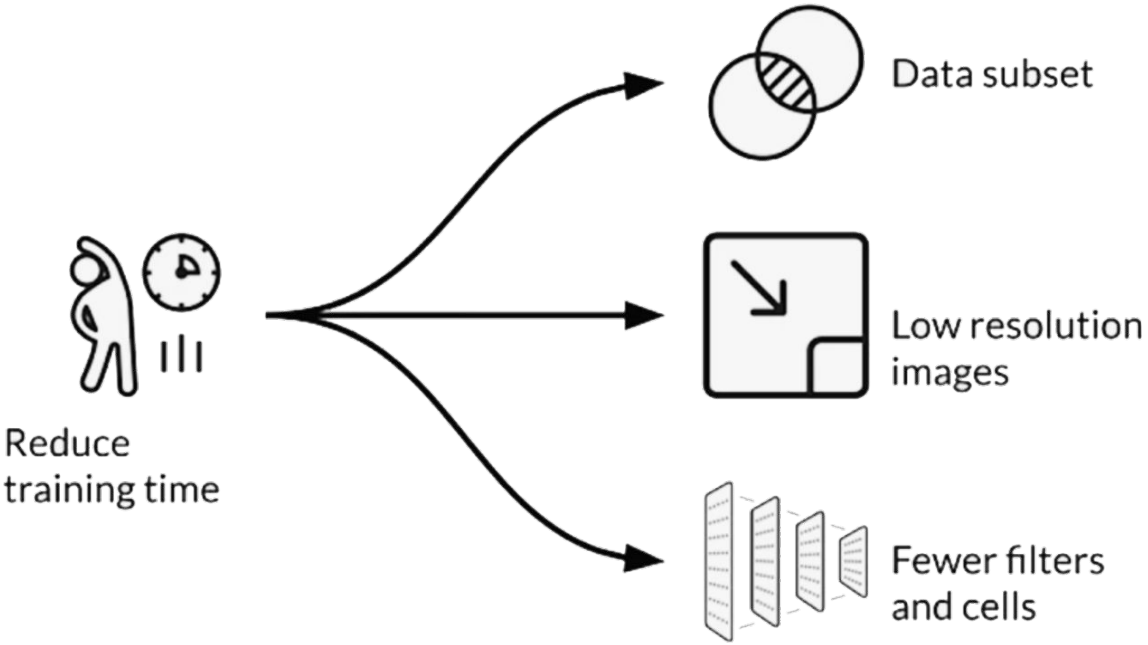
- Grid search
- Random search
- Bayesian optimization
- Evolutionary algorithms
- Reinforcement learning

Performance estimation strategies in NAS

Some strategies are:

- Lower fidelity estimates
- Learning curve extrapolation
- Weight inheritance/Network morphism

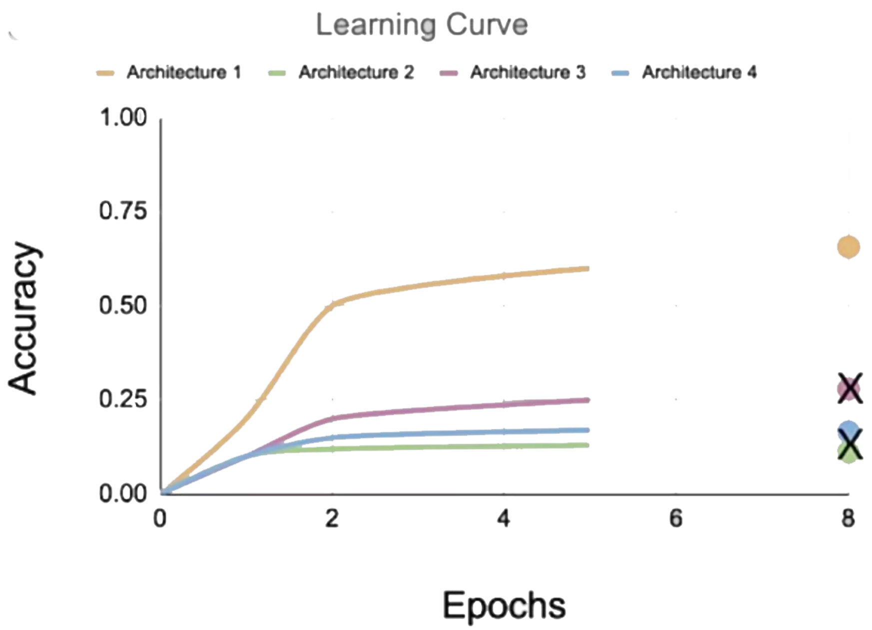
Lower fidelity estimates



- Reduce cost but underestimates performance
- Works if **relative ranking** of architectures does not change due to lower fidelity estimates
- Recent research shows this is not the case

Learning curve extrapolation

- Required predicting learning curves reliably
- Extrapolate based on initial learning
- Removes poor performers



Weight inheritance/Network morphism

- Initializes the weights of novel architectures based on the weights of other architectures that have been trained before
    - similar to transfer learning
- Use network morphism
- Underlying function changed
    - New network Inherits knowledge from past networks
    - Computational speedup: only a few days of GPU
    - Network size not inherently bounded

Machine Learning Systems Design

Modeling Pipeline

Next Lecture: Model Resource Management

CE 40959 Spring 2023
Ali Zarezade
SharifMLSD.github.io

TEMP slides

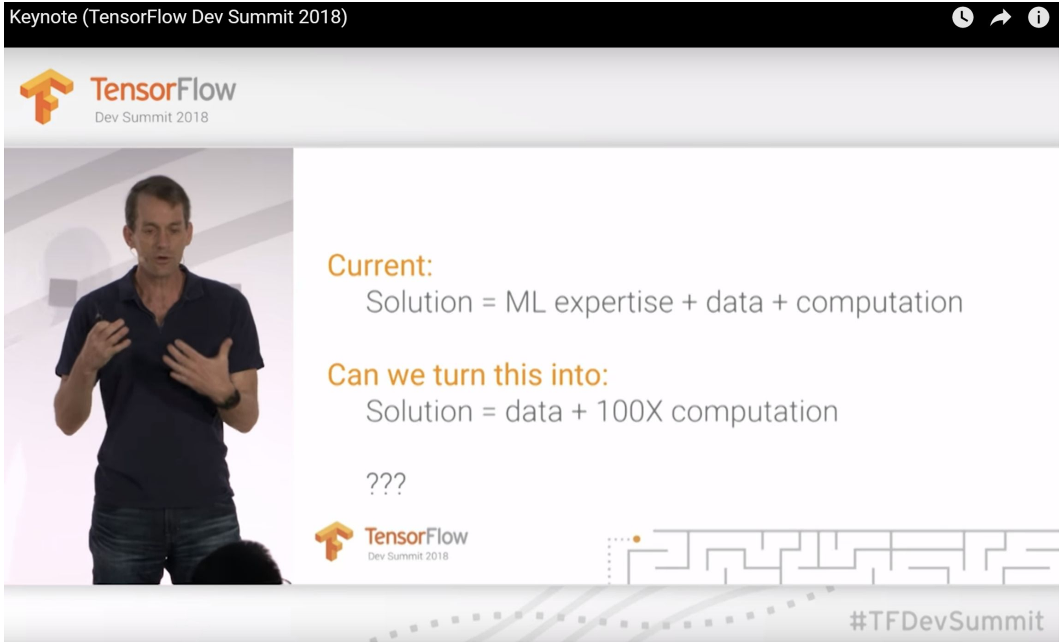
ToDo

Add more HPO method from HPO book

AutoML

AutoML

- A good ML researcher is someone who will automate themselves out of job
- Google: what if we replace ML experts with 100x compute?



AutoML

- Soft AutoML:
  - hyperparameter tuning
- Hard AutoML
  - neural architecture search
  - learned optimizer

More computationally expensive

Soft AutoML: Hyperparameter tuning

- Weaker models with well-tuned hyperparameters can outperform fancier models
    - On the State of the Art of Evaluation in Neural Language Models (Melis et al. 2018)

Soft AutoML: Hyperparameter tuning

- Many hyperparameters to tune

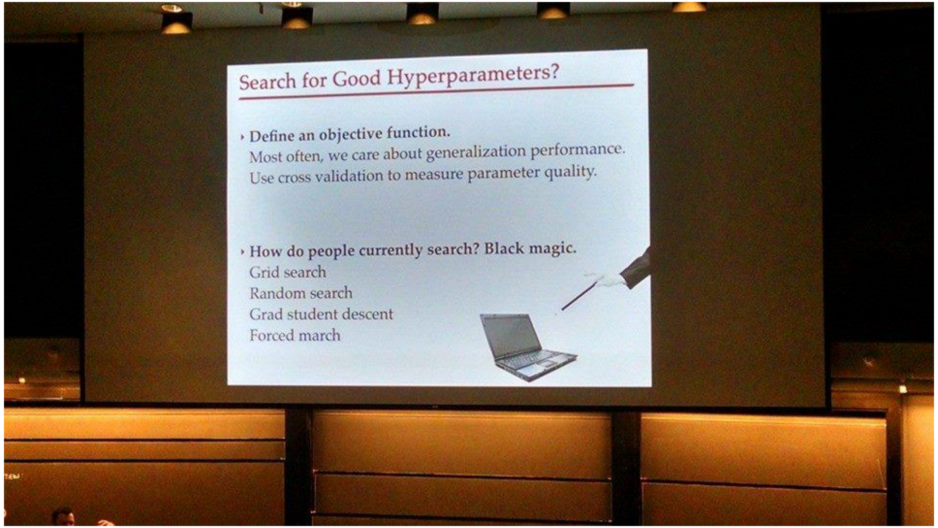
```
model_type = "bert"

def __init__(
    self,
    vocab_size=30522,
    hidden_size=768,
    num_hidden_layers=12,
    num_attention_heads=12,
    intermediate_size=3072,
    hidden_act="gelu",
    hidden_dropout_prob=0.1,
    attention_probs_dropout_prob=0.1,
    max_position_embeddings=512,
    type_vocab_size=2,
    initializer_range=0.02,
    layer_norm_eps=1e-12,
    pad_token_id=0,
    position_embedding_type="absolute",
    use_cache=True,
    classifier_dropout=None,
    **kwargs
):
```

https://github.com/huggingface/transformers/blob/master/src/transformers/models/bert/configuration_bert.py

Soft AutoML: Hyperparameter tuning

- Graduate Student Descent (GSD)
  - A graduate student fiddles around with the hyperparameters until the model works



https://twitter.com/GuyZys/status/592847074170896384/photo/1

Soft AutoML: Hyperparameter tuning

- Hyperparam tuning has become a standard part of ML workflows
- Built-in with frameworks
  - TensorFlow: Keras Turner
  - scikit-learn: auto-sklearn
  - Ray Tune
- Popular algos:
  - Random search
  - Grid search
  - Bayesian optimization

NAS: Neural architecture search

- **Search space**
  - Set of operations
    - e.g. convolution, fully-connected, pooling
  - How operations can be connected

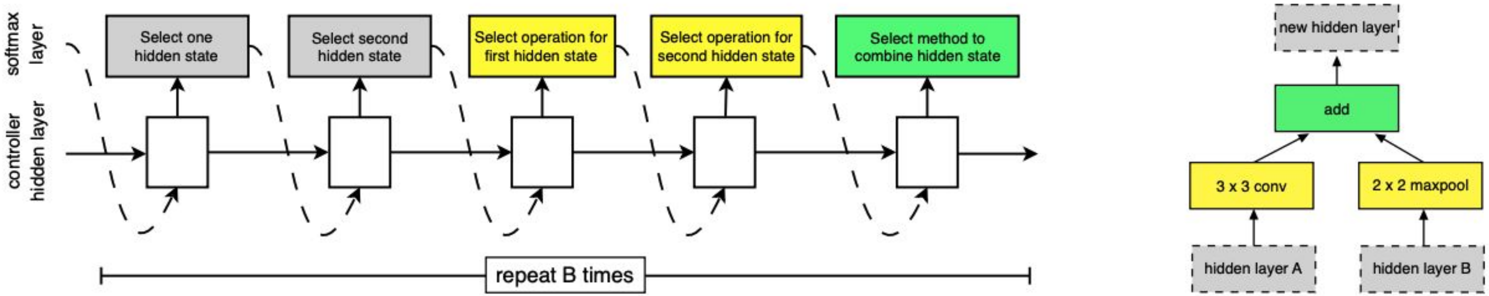


Figure 3. Controller model architecture for recursively constructing one block of a convolutional cell. Each block requires selecting 5 discrete parameters, each of which corresponds to the output of a softmax layer. Example constructed block shown on right. A convolutional cell contains B blocks, hence the controller contains $5B$ softmax layers for predicting the architecture of a convolutional cell. In our experiments, the number of blocks B is 5.

Learning Transferable Architectures for Scalable Image Recognition (Zoph et al., 2017)

NAS: Neural architecture search

- Search space
- **Performance estimation strategy**
  - How to evaluate **many** candidate architectures?
  - Ideal: should be done without having to re-construct or re-train them from scratch.

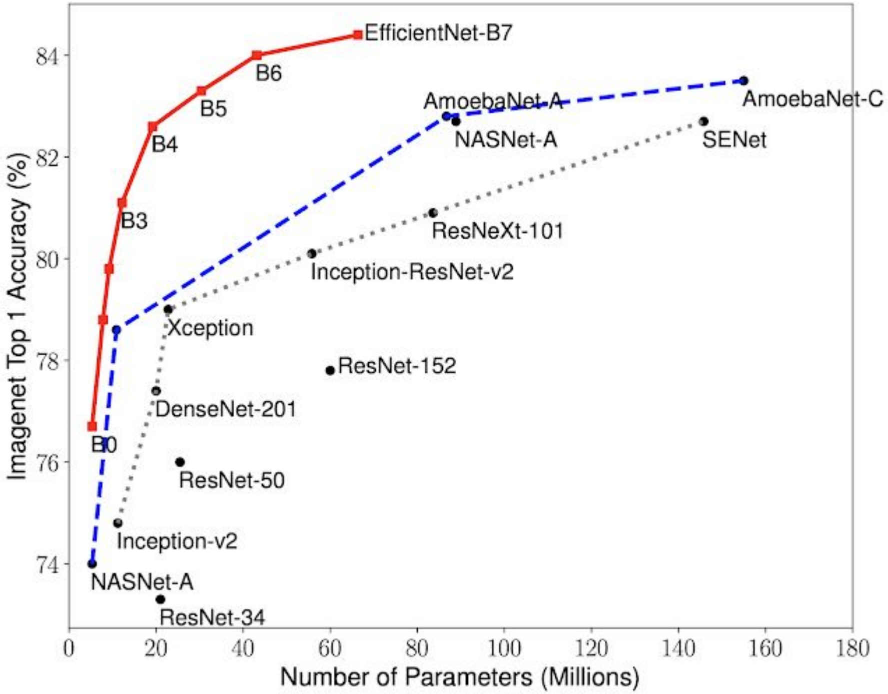
NAS: Neural architecture search

- Search space
- Performance estimation strategy
- **Search strategy**
  - Random
  - Reinforcement learning
    - reward the choices that improve performance estimation
  - Evolution
    - mutate an architecture
    - choose the best-performing offsprings
    - so on

NAS: Neural architecture search

- **Search space**
- **Performance estimation strategy**
- **Search strategy**

Very successful



EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks (Tan et Le., 2019)

Learning: architecture + learning algorithm

- Learning algorithm:
    - A set of functions that specifies how to update the weights.
    - Also called **optimizers**
        - Adam, Momentum, SGD

Learned optimizer

Deep learning

engineering features → learning features

SIFT (Lowe et. al. 1999)
HOG (Dalal et. al. 2005)

LeNet (LeCun et. al. 1998)
AlexNet (Krizhevsky et. al. 2012)

Meta learning

engineering to learn → learning to learn

SGD (Robbins et. al. 1951, Bottou 2010)
Autoencoders (Hinton et. al. 2006)

Learning To Learn (Hochreiter et. al. 2001)
Learned Optimizers (Andrychowicz et. al. 2016, Li et. al. 2016, Wichrowska et. al. 2017, Metz et. al. 2018, 2019)

Slide courtesy of Luke Metz

Learned optimizer

- Learn how to learn on a set of tasks
- Generalize to new tasks

Using a thousand optimization tasks to learn hyperparameter search strategies

Luke Metz [1] **Niru Maheswaranathan** [1] **Ruoxi Sun** [1] **C. Daniel Freeman** [1] **Ben Poole** [1] **Jascha Sohl-Dickstein** [1]

Slide courtesy of Luke Metz

Learned optimizer

- Learn how to learn on a set of tasks
- Generalize to new tasks
- The learned optimizer can then be used to train a better version of itself!

Using a thousand optimization tasks to learn hyperparameter search strategies

Luke Metz [1] Niru Maheswaranathan [1] Ruoxi Sun [1] C. Daniel Freeman [1] Ben Poole [1] Jascha Sohl-Dickstein [1]

Slide courtesy of Luke Metz