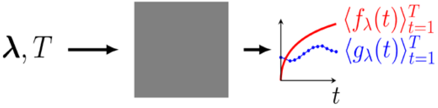
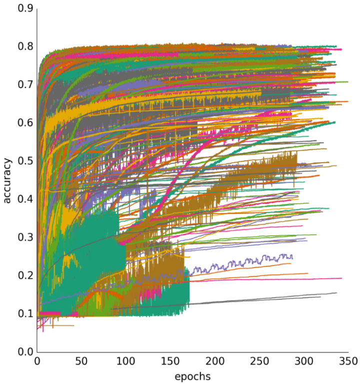
Speedup Techniques for Hyperparameter Optimization

Predicting Learning Curves

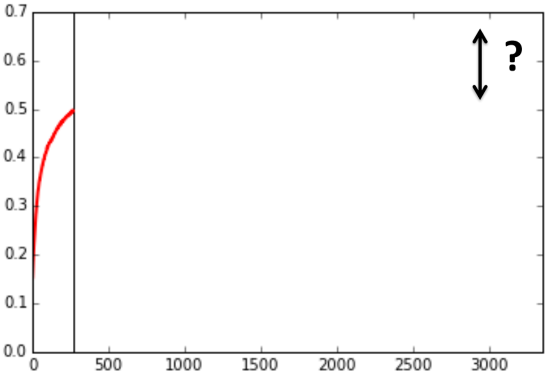
Bernd Bischl   Frank Hutter   Lars Kotthoff
Marius Lindauer   Joaquin Vanschoren

Learning Curves



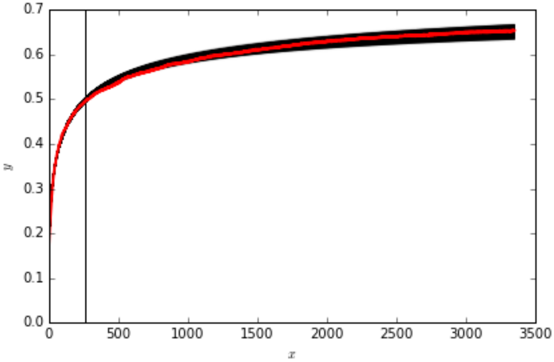


Learning Curve Predictions



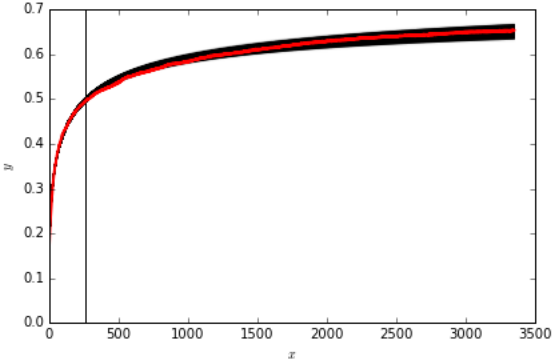
1. Observe learning curve for the first n steps (here $n = 250$)

Learning Curve Predictions



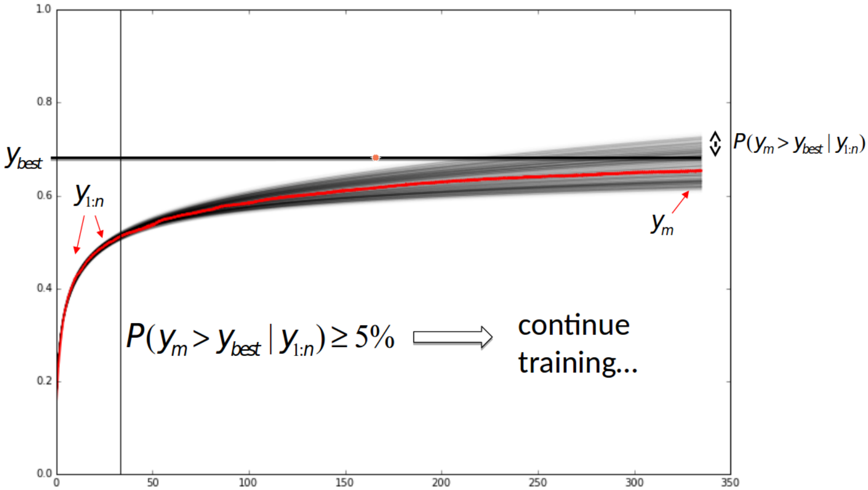
1. Observe learning curve for the first n steps (here $n = 250$)
2. Extrapolation: fit parametric model on partial learning curve to predict remaining learning curve

Learning Curve Predictions



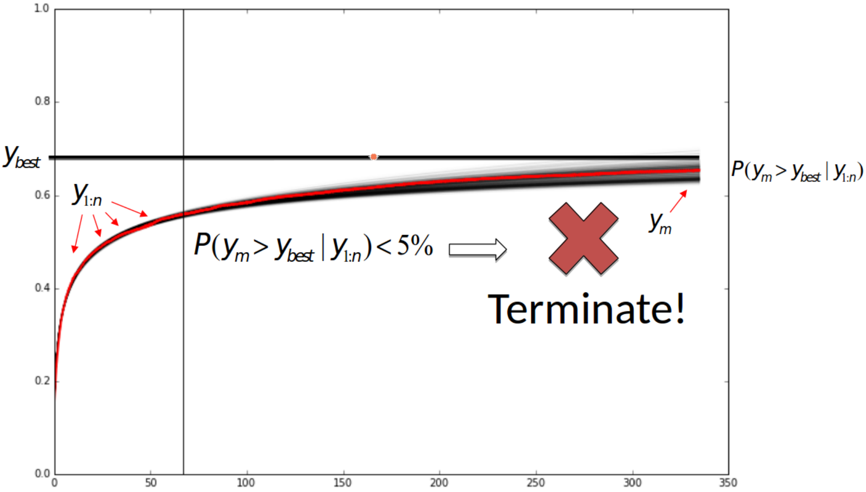
1. Observe learning curve for the first n steps (here $n = 250$)
2. Extrapolation: fit parametric model on partial learning curve to predict remaining learning curve
   - Various models can be used (see following slides)

Learning Curves: Early Termination



$\rightarrow$ need for probabilistic predictions / quantification of uncertainty

Learning Curves: Early Termination



$\rightarrow$ need for probabilistic predictions / quantification of uncertainty

Parametric Learning Curves [Domhan, Springenberg & Hutter, IJCAI 2015]

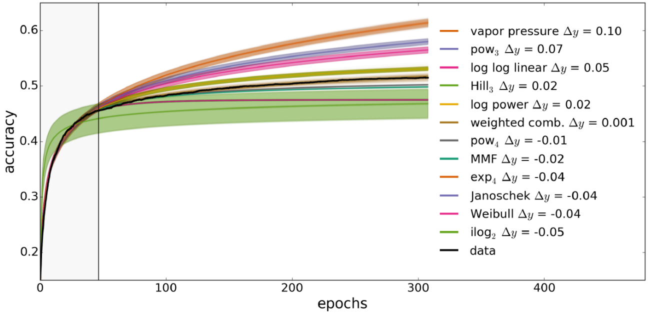
- Use a parametric model f_k with parameters $\boldsymbol{\theta}$ to model performance at step t as: $y_t = f_k(t|\boldsymbol{\theta}) + \epsilon$, with $\epsilon \sim \mathcal{N}(0, \sigma^2)$.

Parametric Learning Curves [Domhan, Springenberg & Hutter, IJCAI 2015]

- Use a parametric model f_k with parameters $\boldsymbol{\theta}$ to model performance at step t as: $y_t = f_k(t|\boldsymbol{\theta}) + \epsilon$, with $\epsilon \sim \mathcal{N}(0, \sigma^2)$.
- Linear combination of $K = 11$ parametric types of models: $f_{comb}(t|\boldsymbol{\xi}) = \sum_{k=1}^{K} w_k f_k(t|\boldsymbol{\theta}_k)$, where $\boldsymbol{\xi} = (w_1, \ldots, w_K, \boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_K, \sigma^2)$

Parametric Learning Curves [Domhan, Springenberg & Hutter, IJCAI 2015]

- Use a parametric model f_k with parameters $\boldsymbol{\theta}$ to model performance at step t as: $y_t = f_k(t|\boldsymbol{\theta}) + \epsilon$, with $\epsilon \sim \mathcal{N}(0, \sigma^2)$.
- Linear combination of $K = 11$ parametric types of models: $f_{comb}(t|\boldsymbol{\xi}) = \sum_{k=1}^{K} w_k f_k(t|\boldsymbol{\theta}_k)$, where $\boldsymbol{\xi} = (w_1, \dots, w_K, \boldsymbol{\theta}_1, \dots, \boldsymbol{\theta}_K, \sigma^2)$

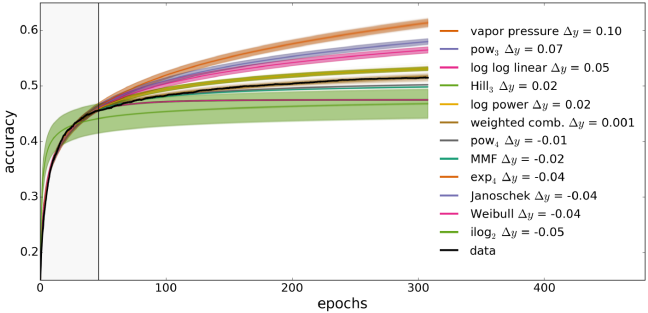


Reference name	Formula
vapor pressure	$\exp(a + \frac{b}{x} + c\log(x))$
pow$_3$	$c - ax^{-\alpha}$
log log linear	$\log(a\log(x) + b)$
Hill$_3$	$\frac{y_{\max} x^\eta}{\kappa^\eta + x^\eta}$
log power	$\frac{a}{1+\left(\frac{x}{e^b}\right)^c}$
pow$_4$	$c - (ax+b)^{-\alpha}$
MMF	$\alpha - \frac{\alpha-\beta}{1+(\kappa x)^\delta}$
exp$_4$	$c - e^{-ax^\alpha + b}$
Janoschek	$\alpha - (\alpha - \beta)e^{-\kappa x^\delta}$
Weibull	$\alpha - (\alpha - \beta)e^{-(\kappa x)^\delta}$
ilog$_2$	$c - \frac{a}{\log x}$

$K = 11$ parametric families for modelling learning curves

Parametric Learning Curves [Domhan, Springenberg & Hutter, IJCAI 2015]

- Use a parametric model f_k with parameters $\boldsymbol{\theta}$ to model performance at step t as: $y_t = f_k(t|\boldsymbol{\theta}) + \epsilon$, with $\epsilon \sim \mathcal{N}(0, \sigma^2)$.
- Linear combination of $K = 11$ parametric types of models: $f_{comb}(t|\boldsymbol{\xi}) = \sum_{k=1}^{K} w_k f_k(t|\boldsymbol{\theta}_k)$, where $\boldsymbol{\xi} = (w_1, \ldots, w_K, \boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_K, \sigma^2)$

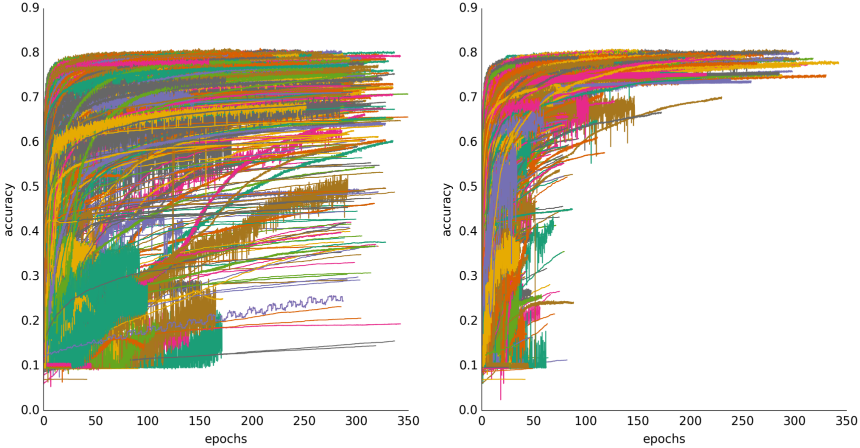


Reference name	Formula
vapor pressure	$\exp(a + \frac{b}{x} + c\log(x))$
pow$_3$	$c - ax^{-\alpha}$
log log linear	$\log(a\log(x) + b)$
Hill$_3$	$\frac{y_{\max} x^\eta}{\kappa^\eta + x^\eta}$
log power	$\frac{a}{1+\left(\frac{x}{e^b}\right)^c}$
pow$_4$	$c - (ax+b)^{-\alpha}$
MMF	$\alpha - \frac{\alpha-\beta}{1+(\kappa x)^\delta}$
exp$_4$	$c - e^{-ax^\alpha + b}$
Janoschek	$\alpha - (\alpha-\beta)e^{-\kappa x^\delta}$
Weibull	$\alpha - (\alpha-\beta)e^{-(\kappa x)^\delta}$
ilog$_2$	$c - \frac{a}{\log x}$

$K = 11$ parametric families for modelling learning curves

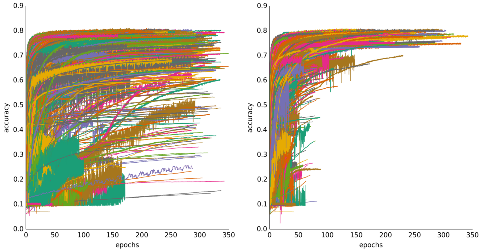
- Use Markov Chain Monte Carlo sampling of $\boldsymbol{\xi}$ to obtain uncertainties

Predictive Termination



All learning curves vs. learning curves with early termination

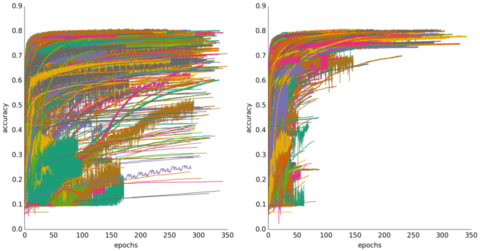
Predictive Termination



All learning curves vs. learning curves with early termination

- Disadvantages of this model?

Predictive Termination



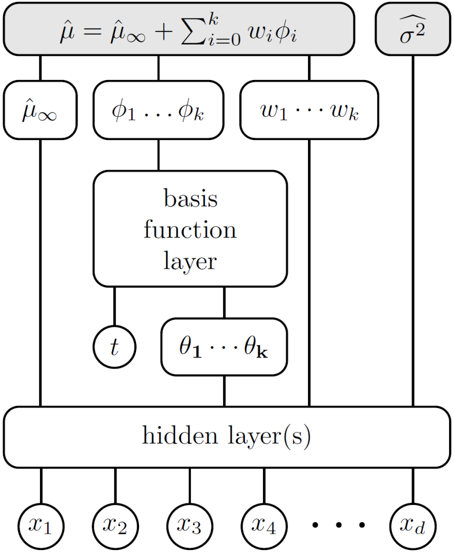
All learning curves vs. learning curves with early termination

- Disadvantages of this model?
  - Relies on manually-selected parametric families of curves
  - Does not take into account hyperparameters used

    $\rightarrow$ can't learn across hyperparameters

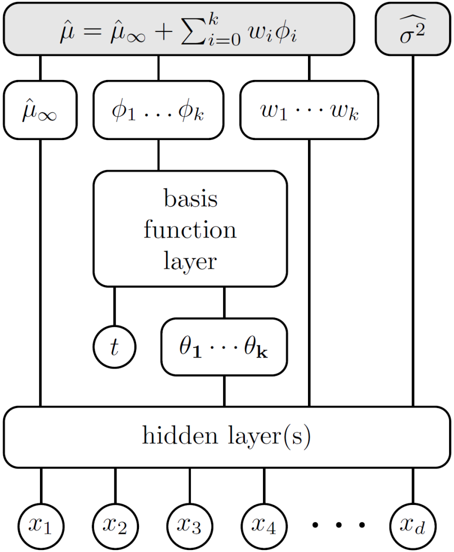
LC-Net [Klein, Falkner & Hutter, ICLR 2017]

- Make a layer out of the parametric learning curves by Domhan et al.
- Also support hyperparameters as inputs (in the figure denoted by $x_1, \ldots, x_d$)



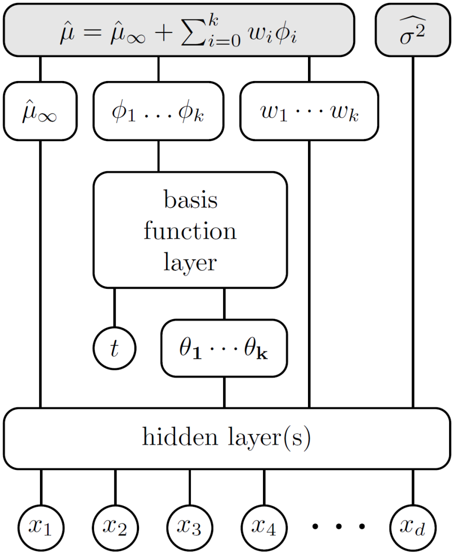
LC-Net [Klein, Falkner & Hutter, ICLR 2017]

- Make a layer out of the parametric learning curves by Domhan et al.
- Also support hyperparameters as inputs (in the figure denoted by $x_1, \ldots, x_d$)
- Disadvantages of this model?



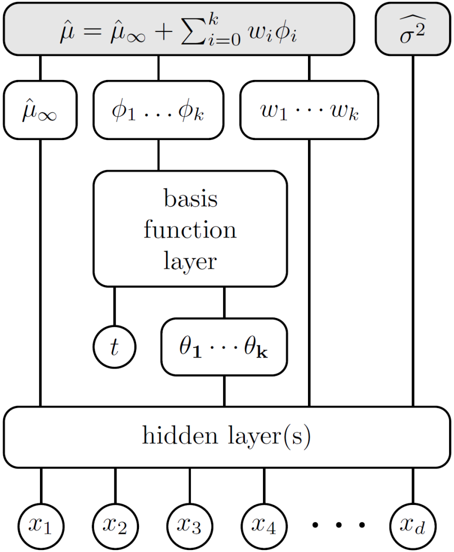
LC-Net [Klein, Falkner & Hutter, ICLR 2017]

- Make a layer out of the parametric learning curves by Domhan et al.
- Also support hyperparameters as inputs (in the figure denoted by $x_1, \dots, x_d$)
- Disadvantages of this model?
  - Relies on manually-selected parametric families of curves
  - Cannot quickly integrate new information from extending the current curve (or from new runs)



LC-Net [Klein, Falkner & Hutter, ICLR 2017]

- Make a layer out of the parametric learning curves by Domhan et al.
- Also support hyperparameters as inputs (in the figure denoted by $x_1, \ldots, x_d$)
- Disadvantages of this model?
  - Relies on manually-selected parametric families of curves
  - Cannot quickly integrate new information from extending the current curve (or from new runs)
  - Also, the model is very hard to train

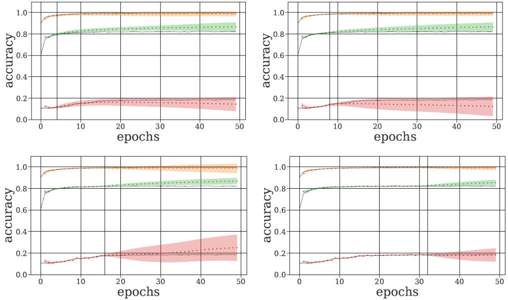


Sequence Models (e.g., Bayesian RNN) [Gargiani et al, AutoML WS 2019]

- Learning curves are sequences
  - Previous models don't treat them like this
  - We can use an RNN (in particular, an LSTM) to predict the next value from a given sequence
  - We can use variational dropout to obtain uncertainty estimates:

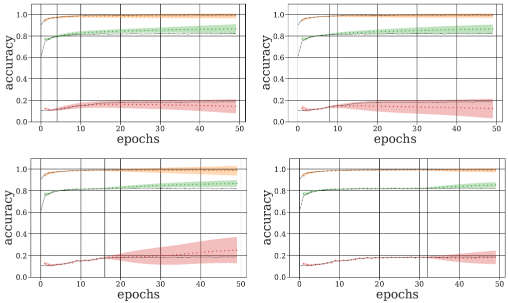
Sequence Models (e.g., Bayesian RNN) [Gargiani et al, AutoML WS 2019]

- Learning curves are sequences
  - Previous models don't treat them like this
  - We can use an RNN (in particular, an LSTM) to predict the next value from a given sequence
  - We can use variational dropout to obtain uncertainty estimates:



Sequence Models (e.g., Bayesian RNN) [Gargiani et al, AutoML WS 2019]

- Learning curves are sequences
  - Previous models don't treat them like this
  - We can use an RNN (in particular, an LSTM) to predict the next value from a given sequence
  - We can use variational dropout to obtain uncertainty estimates:



Note: we can also use a simpler model

- E.g., a random forest to map from a fixed-size window to the next value

Compare: Baker et al, 2017 [Baker et al, ICLR WS 2018]

- Idea: map from configurations (including architectural hyperparameters) and partial learning curves to the final performance
- Advantages
  - Much simpler idea than all the approaches just discussed: no need to model the entire learning curve
  - Much easier to implement
- Disadvantage?

Compare: Baker et al, 2017 [Baker et al, ICLR WS 2018]

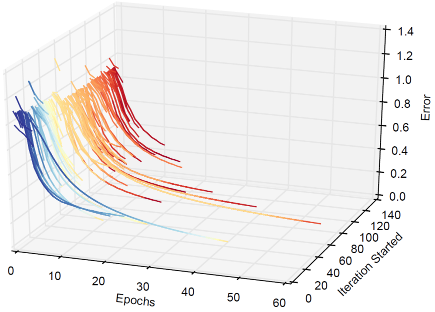
- Idea: map from configurations (including architectural hyperparameters) and partial learning curves to the final performance
- Advantages
  - Much simpler idea than all the approaches just discussed: no need to model the entire learning curve
  - Much easier to implement
- Disadvantage? $\rightarrow$ requires many (e.g., 100) fully-evaluated learning curves as training data
  - After 100 full function evaluations we want to be pretty much converged in practice
  - But definitely helpful for speeding up RL

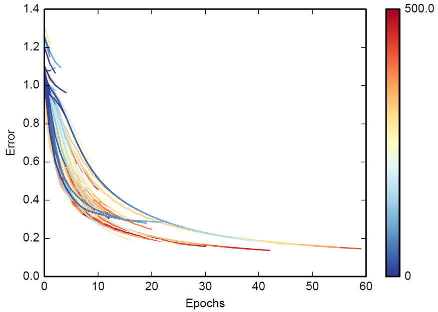
Freeze-Thaw Bayesian Optimization [Swersky et al, arXiv 2014]

- Use a Gaussian process with inputs $\boldsymbol{\lambda}$ and t; special kernel for t
- For N configurations and T epochs each: $O(N^3 t^3) \rightarrow$ approximation
- Iteratively: either extend existing configuration or try new one

Freeze-Thaw Bayesian Optimization [Swersky et al, arXiv 2014]

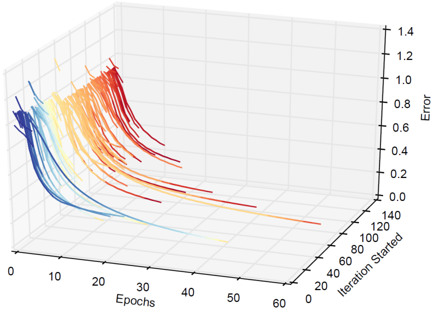
- Use a Gaussian process with inputs $\boldsymbol{\lambda}$ and t; special kernel for t
- For N configurations and T epochs each: $O(N^3 t^3) \rightarrow$ approximation
- Iteratively: either extend existing configuration or try new one
- Result for probabilistic matrix factorization:

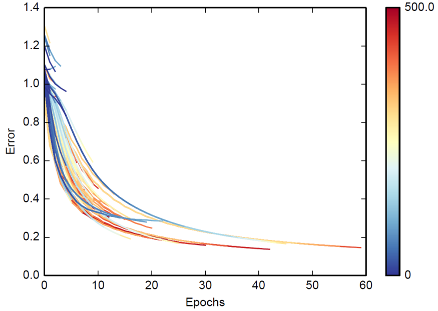




Freeze-Thaw Bayesian Optimization [Swersky et al, arXiv 2014]

- Use a Gaussian process with inputs $\boldsymbol{\lambda}$ and t; special kernel for t
- For N configurations and T epochs each: $O(N^3 t^3) \rightarrow$ approximation
- Iteratively: either extend existing configuration or try new one
- Result for probabilistic matrix factorization:





- Unfortunately, no results for DNNs; no code available

Questions to Answer for Yourself / Discuss with Friends

- Repetition. List all learning curve prediction methods you recall, along with their pros and cons.
- Discussion. Could predictive termination cut off evaluations early that would turn out to be the best?
- Discussion. How would you determine a learning curve prediction method's own hyperparameters (such as the 5% for early learning curve termination), in practice?
- Discussion. How could we exploit additional side information we gain about the learning curve, such as, e.g., statistics for the size of the gradients and activations over time?