Regularization strategies for Deep Learning

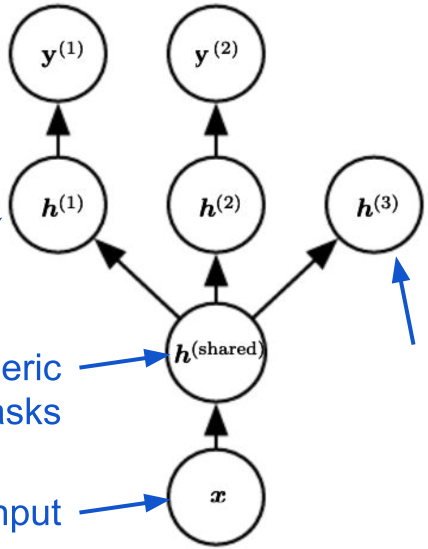
Parameter Sharing, Early Stopping, and Ensemble Learning

N. Rich Nguyen, PhD
SYS 6016

3. Multitask Learning

Multitask Learning / Parameter Sharing

- **Multitask learning** is a way to improve generalization by pooling the examples arising out of several tasks
- Better generalization because of the **shared parameters**, for which statistical strength can be greatly improved comparing to a single task.



Upper layer: Task-specific parameters

Lower layer of representation capturing the generic parameters shared across tasks

Different supervised tasks share the same input

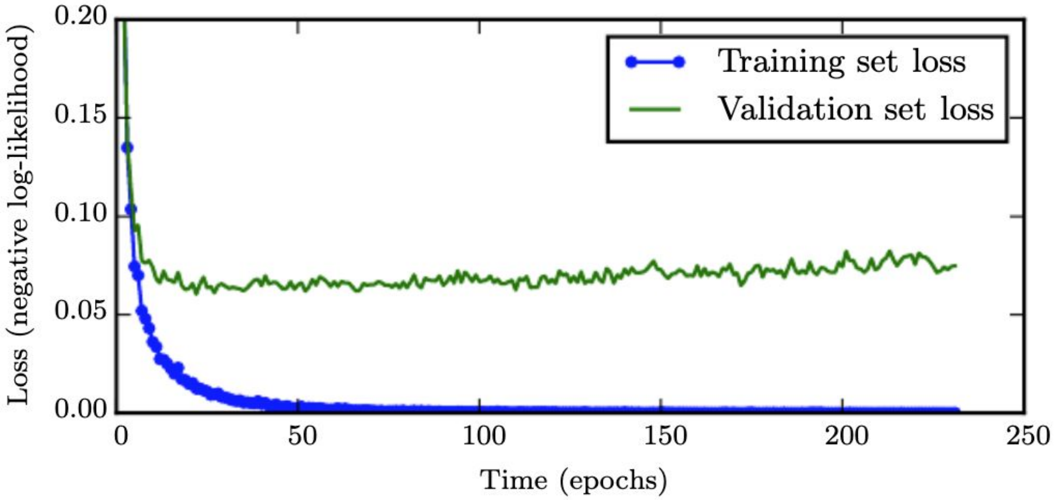
Some upper layer parameters associated with no output tasks, but still explain some of the input variations (used for unsupervised learning)

4. Early Stopping

Early Stopping

When training large models with sufficient capacity to overfit the task, we often observe reliably that training error decreases steadily over time, but validation error begins to rise again → we can stop at the **lowest validation error**!

The **most commonly used form** of regularization in deep learning: simple yet effective!



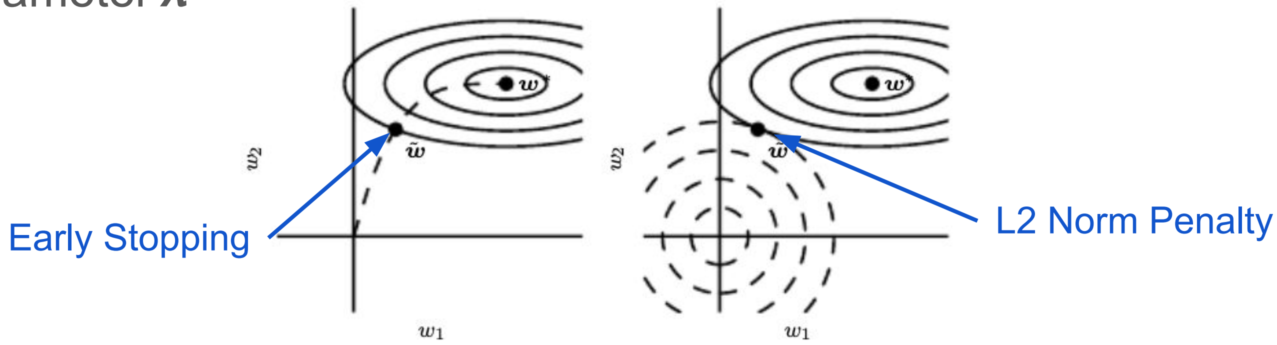
Early Stopping: Algorithm

Algorithm 7.1 The early stopping meta-algorithm for determining the best amount of time to train. This meta-algorithm is a general strategy that works well with a variety of training algorithms and ways of quantifying error on the validation set.

Let n be the number of steps between evaluations.
Let p be the "patience," the number of times to observe worsening validation set error before giving up.
Let $\boldsymbol{\theta}_o$ be the initial parameters.
$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta}_o$
$i \leftarrow 0$
$j \leftarrow 0$
$v \leftarrow \infty$
$\boldsymbol{\theta}^* \leftarrow \boldsymbol{\theta}$
$i^* \leftarrow i$
while $j < p$ **do**
  Update $\boldsymbol{\theta}$ by running the training algorithm for n steps.
  $i \leftarrow i + n$
  $v' \leftarrow \text{ValidationSetError}(\boldsymbol{\theta})$
  if $v' < v$ **then**
    $j \leftarrow 0$
    $\boldsymbol{\theta}^* \leftarrow \boldsymbol{\theta}$
    $i^* \leftarrow i$
    $v \leftarrow v'$
  else
    $j \leftarrow j + 1$
  end if
end while
Best parameters are $\boldsymbol{\theta}^*$, best number of training steps is i^*.

Early Stopping: Properties

- Is a very efficient **hyperparameter selection** algorithm (saves the best set).
- Controls the **model capacity** by determining how many steps to fit the training data (and reduce the **computational cost** of the training procedure)
- **Unobtrusive** form of regularization as almost no change in training procedure
- Can be used either **alone or with** other regularization strategies
- Is **equivalent** to L^2 regularization, and **better** because it automatically determines the correct amount of regularization while L^2 requires the tuning of hyperparameter $\boldsymbol{\lambda}$



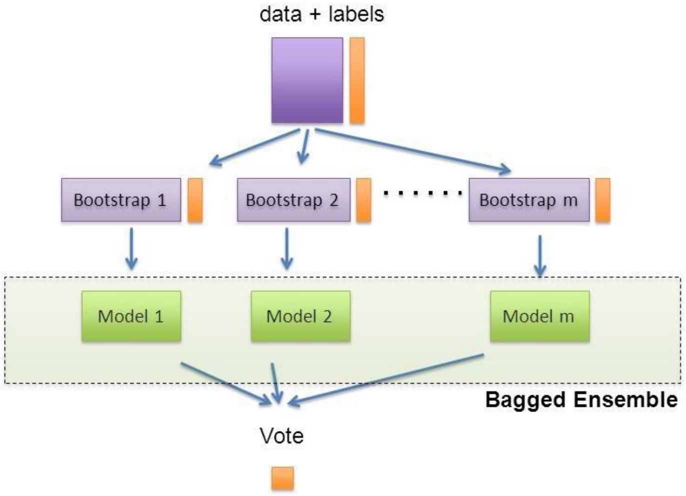
5. Ensemble Learning

Bagging

Bagging (bootstrap aggregating) is a technique for reducing generalization error by combining several models.

- **The idea:** train several models separately, then have all the models vote on the output for test examples.
- **The reason it works:** different models will usually not make all the same errors on the test set.

Techniques employing this idea are known as **ensemble learning**.



Why averaging works?

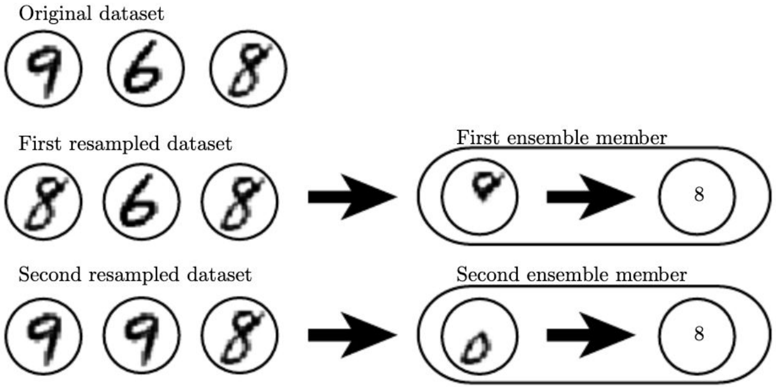
Consider a set of `k` regression models. Each model makes an error ϵ_i on each example, with the errors drawn from a normal distribution with variance $v = \mathbb{E}[\epsilon_i^2]$ and covariance $c = \mathbb{E}[\epsilon_i \epsilon_j]$. The expected square error of the ensemble is

$$\mathbb{E}\left[\left(\frac{1}{k}\sum_i \epsilon_i\right)^2\right] = \frac{1}{k^2}\mathbb{E}\left[\sum_i \left(\epsilon_i^2 + \sum_{j\neq i}\epsilon_i\epsilon_j\right)\right]$$
$$= \frac{1}{k}v + \frac{k-1}{k}c.$$

- When errors of the members are **perfectly correlated** `(c=v)`, this expected square error reduces to `v`, and the ensemble does not help at all
- When errors are **perfectly uncorrected** `(c=0)`, expected square error reduces to `1/k v`, so ensemble error decreases linearly with number of members
- Generally, if the members make independent errors, ensemble does better

Bagging in Neural Networks

- Neural networks often **benefit from model averaging** even if all of the models are trained on the same dataset.
- Differences in **random weight initialization**, random selection of the training **minibatches**, in **hyperparameters** → cause different members of the ensemble to make independence errors.
- Modeling averaging is powerful and reliable, but is *discouraged* when **benchmarking** algorithms for scientific papers.



“Our winning model is an ensemble of 107 models. Our experience is that most efforts should be concentrated in deriving substantially different approaches, rather than refining a simple technique.”