Regularization I

Parameter regularization

REGULARIZATION

Topics: Why we regularize.

- How do we ensure our model will perform well not just on the training data, but also on new inputs?
  - i.e. How do we ensure **generalization**?
- Our main strategy: **Regularization**
- Definition: Putting extra constraints on a machine learning model, to improved performance on the **test set** (not training set), either by encoding prior knowledge into the model, or by forcing the model to consider alternative hypotheses that explain the training data.

REGULARIZATION

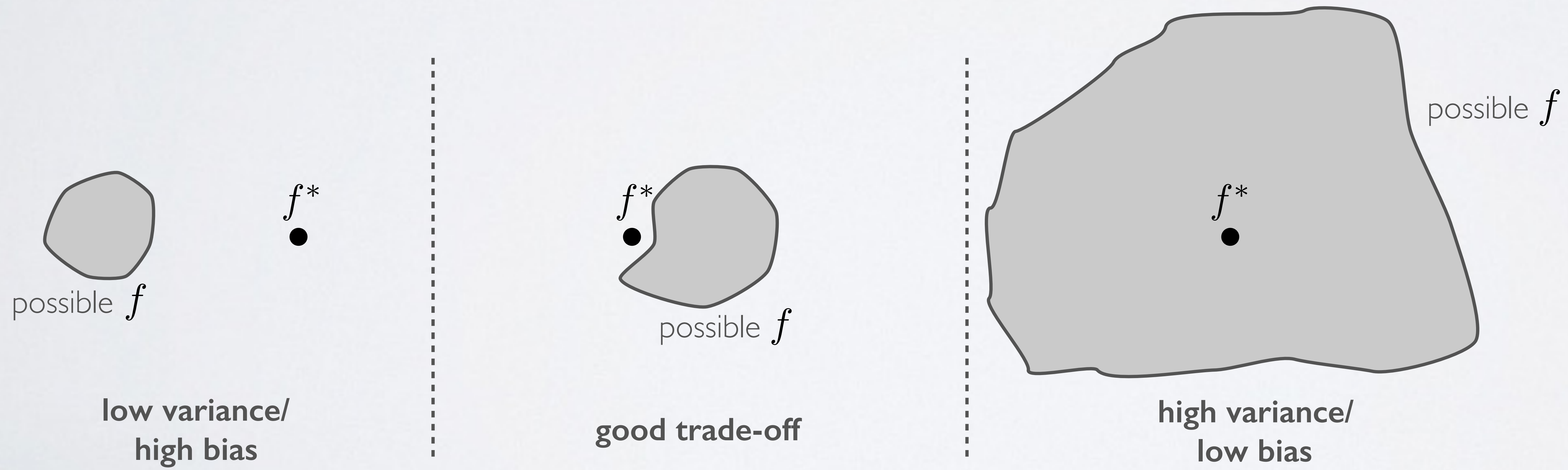
Topics: Why we regularize.

- Regularizers trade increased bias for reduced variance.
- An effective regularizer is one that makes a profitable trade, that is, it reduces variance significantly while not overly increasing the bias.

REGULARIZATION

Topics: bias-variance trade-off

- Variance of trained model: does it vary a lot if the training set changes
- Bias of trained model: is the average model close to the true solution
- Generalization error can be seen as the sum of the (squared) bias and the variance (the mean squared error - MSE)



REGULARIZATION

In the context of deep learning:

- An overly complex model family does not necessarily include (or even come close to) the target function or the true data generating process.
- Often working with data such as images, audio sequences and text:
  - we can safely assume that the model family we are training does not include the data generating process.
- Consequence: complexity of the model is not about finding the model of the right size. i.e. the right number of parameters.
- Instead, the best fitting model is one that possesses a large number of parameters that are not entirely free to span their domain.

REGULARIZATION

Recall: **Structured risk** (loss function we are minimizing)

$$\tilde{J}(\boldsymbol{\theta}; \boldsymbol{X}, \boldsymbol{y}) = \underbrace{J(\boldsymbol{\theta}; \boldsymbol{X}, \boldsymbol{y})}_{\text{Empirical Risk}} + \alpha \underbrace{\Omega(\boldsymbol{\theta})}_{\text{Regularization Term}}$$

- α is a hyperparameter that balances the relative effect of the regularization term.
- For neural network models, we typically have $\boldsymbol{\theta} = [\boldsymbol{w}^\top, b]^\top$
- Typical regularizers: $\Omega(\boldsymbol{\theta}) = \frac{1}{2}||\boldsymbol{w}||_2^2$ or $\Omega(\boldsymbol{\theta}) = ||\boldsymbol{w}||_1$

REGULARIZATION

Topics: L2 regularization

$$\Omega(\boldsymbol{\theta}) = \frac{1}{2}||\boldsymbol{w}||_2^2$$

$$\tilde{J}(\boldsymbol{\theta}; \boldsymbol{X}, \boldsymbol{y}) = J(\boldsymbol{\theta}; \boldsymbol{X}, \boldsymbol{y}) + \alpha\Omega(\boldsymbol{\theta})$$

Taking gradient

$$\nabla_{\boldsymbol{w}} \tilde{J}(\boldsymbol{w}; \boldsymbol{X}, \boldsymbol{y}) = \alpha \boldsymbol{w} + \nabla_{\boldsymbol{w}} J(\boldsymbol{X}, \boldsymbol{y}; \boldsymbol{w})$$

We are going to gain some insight into how this works!

REGULARIZATION

Topics: L2 regularization $\Omega(\boldsymbol{\theta}) = \frac{1}{2}||\boldsymbol{w}||_2^2$

$$\tilde{J}(\boldsymbol{\theta}; \boldsymbol{X}, \boldsymbol{y}) = J(\boldsymbol{\theta}; \boldsymbol{X}, \boldsymbol{y}) + \alpha\Omega(\boldsymbol{\theta})$$

- Consider a quadratic approximation to the loss function in the neighbourhood of the empirically optimal value of the weights $\boldsymbol{w}^*$

$$\hat{J}(\boldsymbol{\theta}) = J(\boldsymbol{w}^*) + \frac{1}{2}(\boldsymbol{w} - \boldsymbol{w}^*)^\top \boldsymbol{H}(\boldsymbol{w} - \boldsymbol{w}^*)$$

Taking gradient

$$\nabla_{\boldsymbol{w}} \hat{J}(\boldsymbol{w}) = \boldsymbol{H}(\boldsymbol{w} - \boldsymbol{w}^*)$$

REGULARIZATION

Topics: L2 regularization $\Omega(\boldsymbol{\theta}) = \frac{1}{2}||\boldsymbol{w}||_2^2$

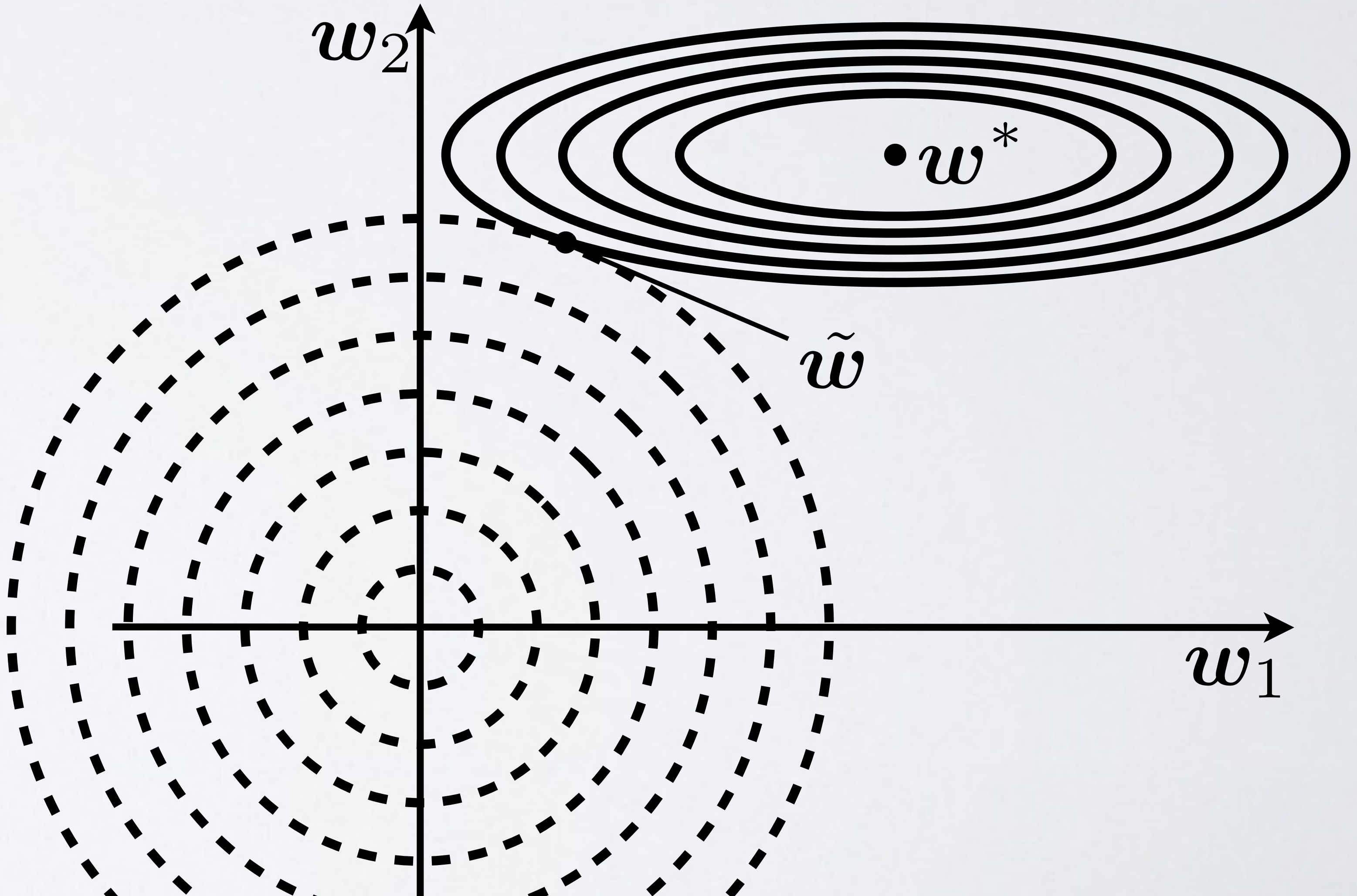
$$\nabla_{\boldsymbol{w}} \hat{J}(\boldsymbol{w}) = \boldsymbol{H}(\boldsymbol{w} - \boldsymbol{w}^*)$$

- Consider the effect of L2 regularization around the unregularized optimum $\boldsymbol{w}^*$

$$\alpha \boldsymbol{w} + \boldsymbol{H}(\boldsymbol{w} - \boldsymbol{w}^*) = 0$$

$$(\boldsymbol{H} + \alpha \boldsymbol{I})\boldsymbol{w} = \boldsymbol{H}\boldsymbol{w}^*$$

$$\tilde{\boldsymbol{w}} = (\boldsymbol{H} + \alpha \boldsymbol{I})^{-1}\boldsymbol{H}\boldsymbol{w}^*$$



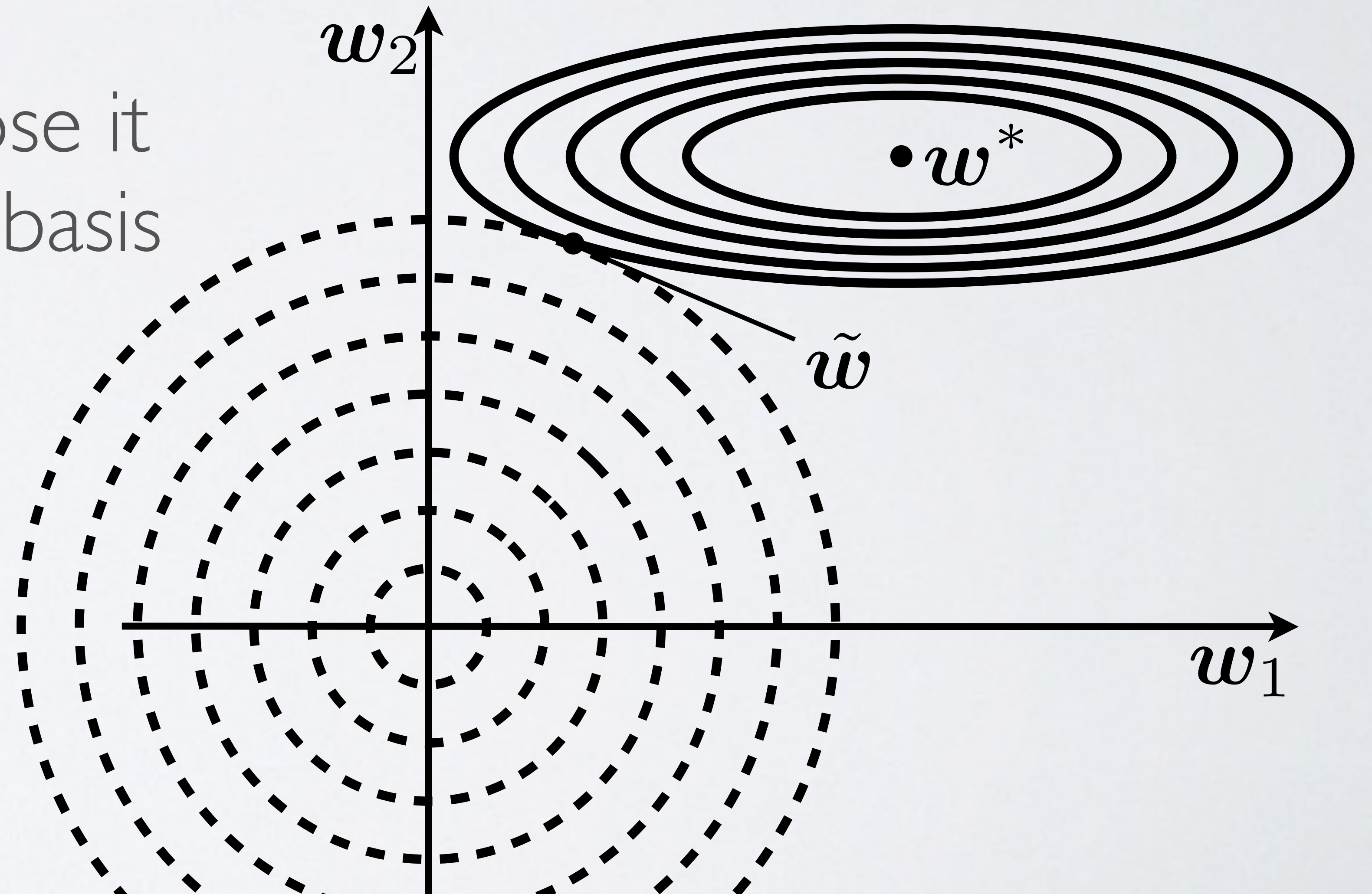
REGULARIZATION

Topics: L2 regularization $\Omega(\boldsymbol{\theta}) = \frac{1}{2}||\boldsymbol{w}||_2^2$

$$\tilde{\boldsymbol{w}} = (\boldsymbol{H} + \alpha \boldsymbol{I})^{-1} \boldsymbol{H} \boldsymbol{w}^*$$

- The presence of the regularization term moves the optimum from $\boldsymbol{w}^*$ to $\tilde{\boldsymbol{w}}$
- $\boldsymbol{H}$ is real and symmetric, so we can decompose it into a diagonal matrix $\boldsymbol{\Lambda}$ and an orthogonal basis of eigenvectors, $\boldsymbol{Q}$, such that:

$$\boldsymbol{H} = \boldsymbol{Q} \boldsymbol{\Lambda} \boldsymbol{Q}^\top$$



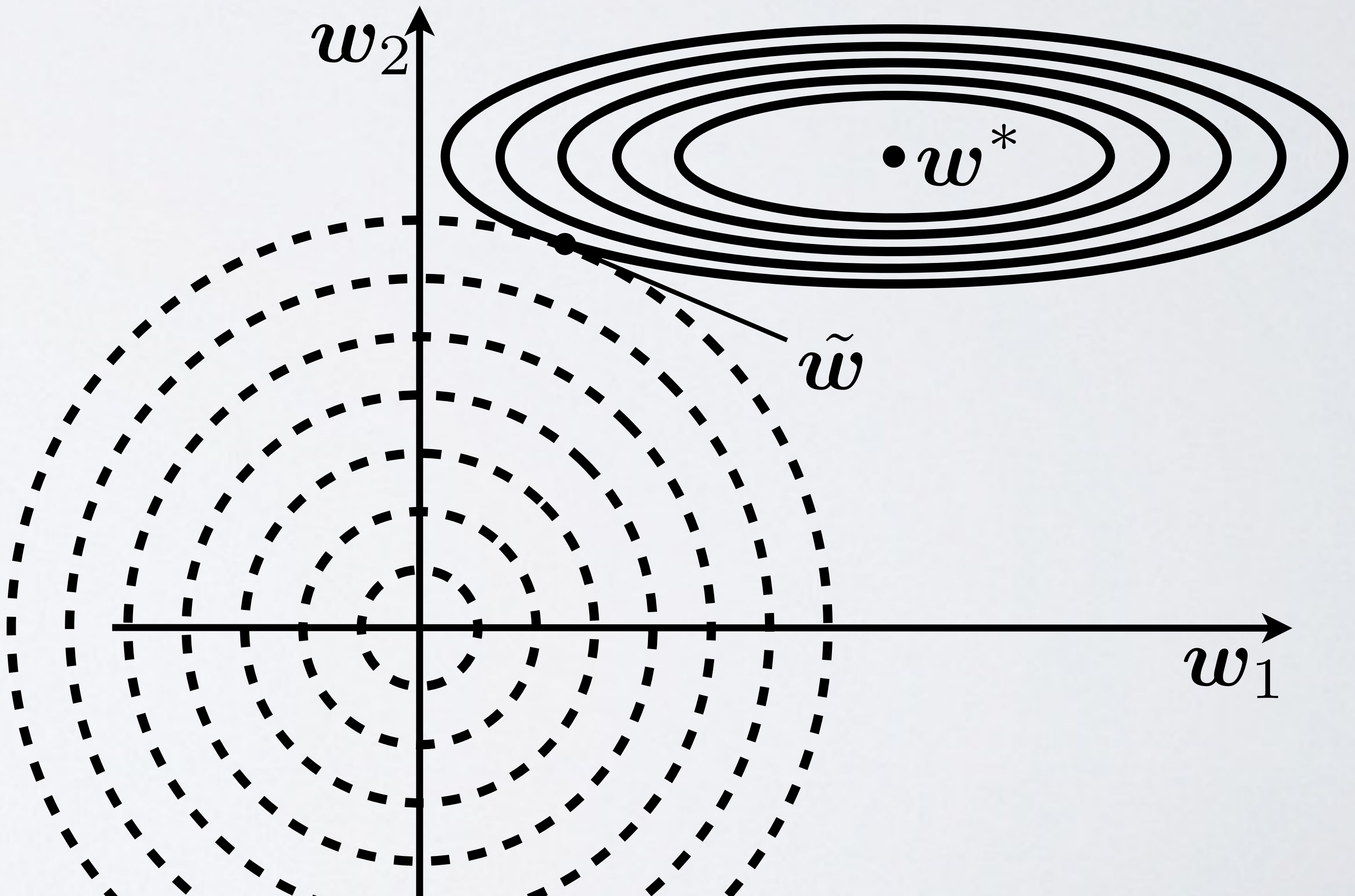
REGULARIZATION

Topics: L2 regularization $\Omega(\boldsymbol{\theta}) = \frac{1}{2}||\boldsymbol{w}||_2^2$

$$\tilde{\boldsymbol{w}} = (\boldsymbol{H} + \alpha \boldsymbol{I})^{-1} \boldsymbol{H} \boldsymbol{w}^*$$

- In the above, if we swap in $\boldsymbol{H} = \boldsymbol{Q} \boldsymbol{\Lambda} \boldsymbol{Q}^\top$ we get:

$$\boldsymbol{Q}^\top \tilde{\boldsymbol{w}} = (\boldsymbol{\Lambda} + \alpha \boldsymbol{I})^{-1} \boldsymbol{\Lambda} \boldsymbol{Q}^\top \boldsymbol{w}^*$$



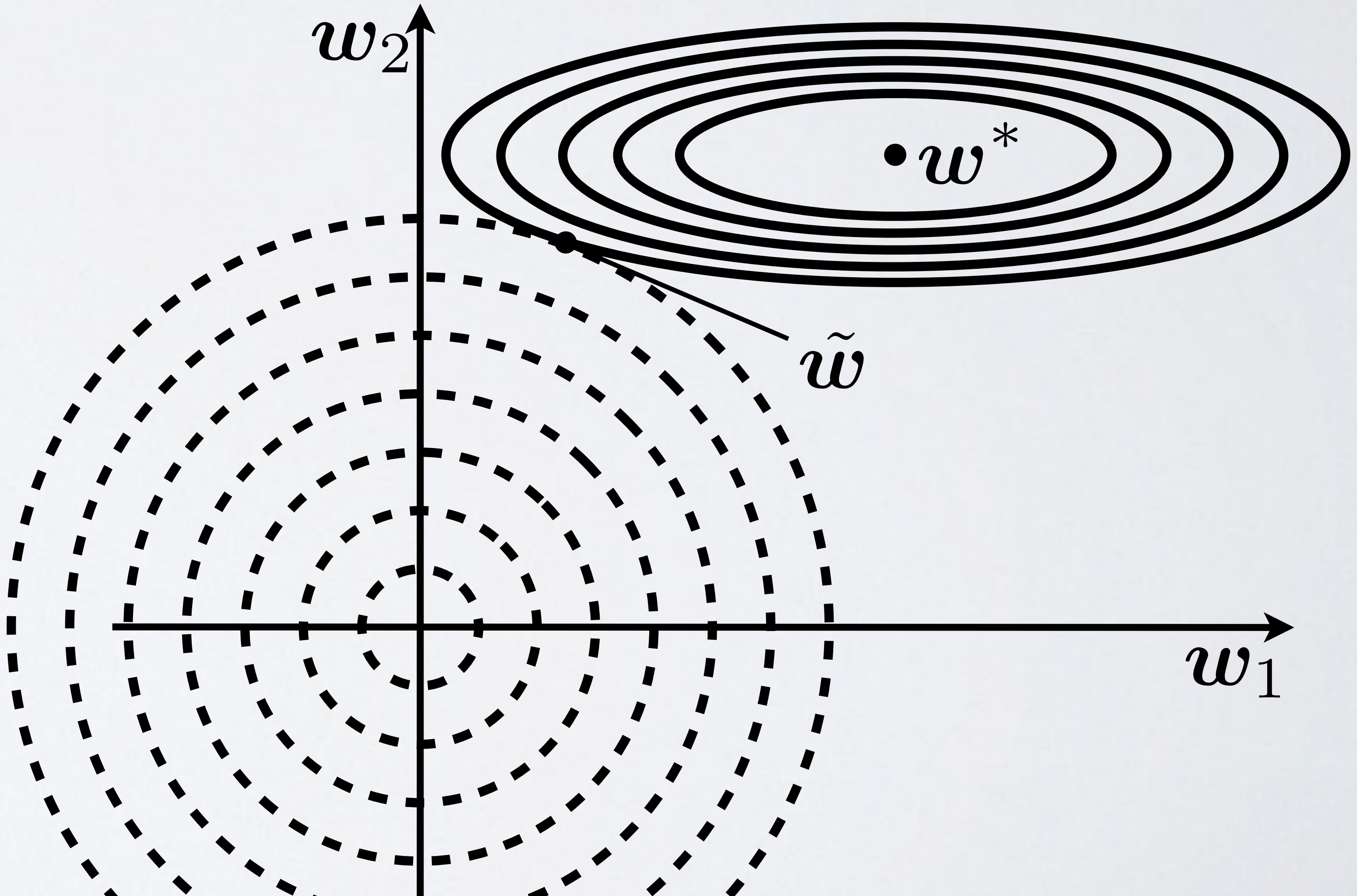
REGULARIZATION

Topics: L2 regularization $\Omega(\boldsymbol{\theta}) = \frac{1}{2}||\boldsymbol{w}||_2^2$

$$\boldsymbol{Q}^\top \tilde{\boldsymbol{w}} = (\boldsymbol{\Lambda} + \alpha \boldsymbol{I})^{-1} \boldsymbol{\Lambda} \boldsymbol{Q}^\top \boldsymbol{w}^*$$

- The different components of $\boldsymbol{w}^*$ are rescaled by the regularization.
- The component aligned with eigenvector i is rescaled by a factor

$$\frac{\lambda_i}{\lambda_i + \alpha}$$



REGULARIZATION

Topics: L1 regularization $\Omega(\boldsymbol{\theta}) = ||\boldsymbol{w}||_1$

$$\tilde{J}(\boldsymbol{\theta}; \boldsymbol{X}, \boldsymbol{y}) = J(\boldsymbol{\theta}; \boldsymbol{X}, \boldsymbol{y}) + \beta\Omega(\boldsymbol{\theta})$$

Taking gradient

$$\nabla_{\boldsymbol{w}}\tilde{J}(\boldsymbol{w}; \boldsymbol{X}, \boldsymbol{y}) = \nabla_{\boldsymbol{w}}J(\boldsymbol{X}, \boldsymbol{y}; \boldsymbol{w}) + \beta \mathrm{sign}(\boldsymbol{w})$$

REGULARIZATION

Topics: L1 regularization $\Omega(\boldsymbol{\theta}) = ||\boldsymbol{w}||_1$

$$\tilde{J}(\boldsymbol{\theta}; \boldsymbol{X}, \boldsymbol{y}) = J(\boldsymbol{\theta}; \boldsymbol{X}, \boldsymbol{y}) + \beta\Omega(\boldsymbol{\theta})$$

- Consider a quadratic approximation to the loss function in the neighbourhood of the empirically optimal value of the weights $\boldsymbol{w}^*$

$$\hat{J}(\boldsymbol{\theta}) = J(\boldsymbol{w}^*) + \frac{1}{2}(\boldsymbol{w} - \boldsymbol{w}^*)^\top \boldsymbol{H}(\boldsymbol{w} - \boldsymbol{w}^*)$$

Taking gradient

$$\nabla_{\boldsymbol{w}} \hat{J}(\boldsymbol{w}) = \boldsymbol{H}(\boldsymbol{w} - \boldsymbol{w}^*)$$

REGULARIZATION

Topics: L1 regularization $\Omega(\boldsymbol{\theta}) = ||\boldsymbol{w}||_1$

$$\hat{J}(\boldsymbol{\theta}) = J(\boldsymbol{w}^*) + \frac{1}{2}(\boldsymbol{w} - \boldsymbol{w}^*)^\top \boldsymbol{H}(\boldsymbol{w} - \boldsymbol{w}^*)$$

Taking gradient

$$\nabla_{\boldsymbol{w}} \hat{J}(\boldsymbol{w}) = \boldsymbol{H}(\boldsymbol{w} - \boldsymbol{w}^*)$$

- We will also make the further simplifying assumption that the Hessian is diagonal, $\boldsymbol{H} = \mathrm{diag}([\gamma_1, \ldots, \gamma_N])$, where each $\gamma_i > 0$

REGULARIZATION

Topics: L1 regularization $\Omega(\boldsymbol{\theta}) = ||\boldsymbol{w}||_1$

- Under these assumptions the objective simplifies to a system of equations:

$$\tilde{J}(\boldsymbol{w}_i; \boldsymbol{X}, \boldsymbol{y}) = \frac{1}{2}\gamma_i(\boldsymbol{w}_i - \boldsymbol{w}_i^*)^2 + \beta|\boldsymbol{w}_i|.$$

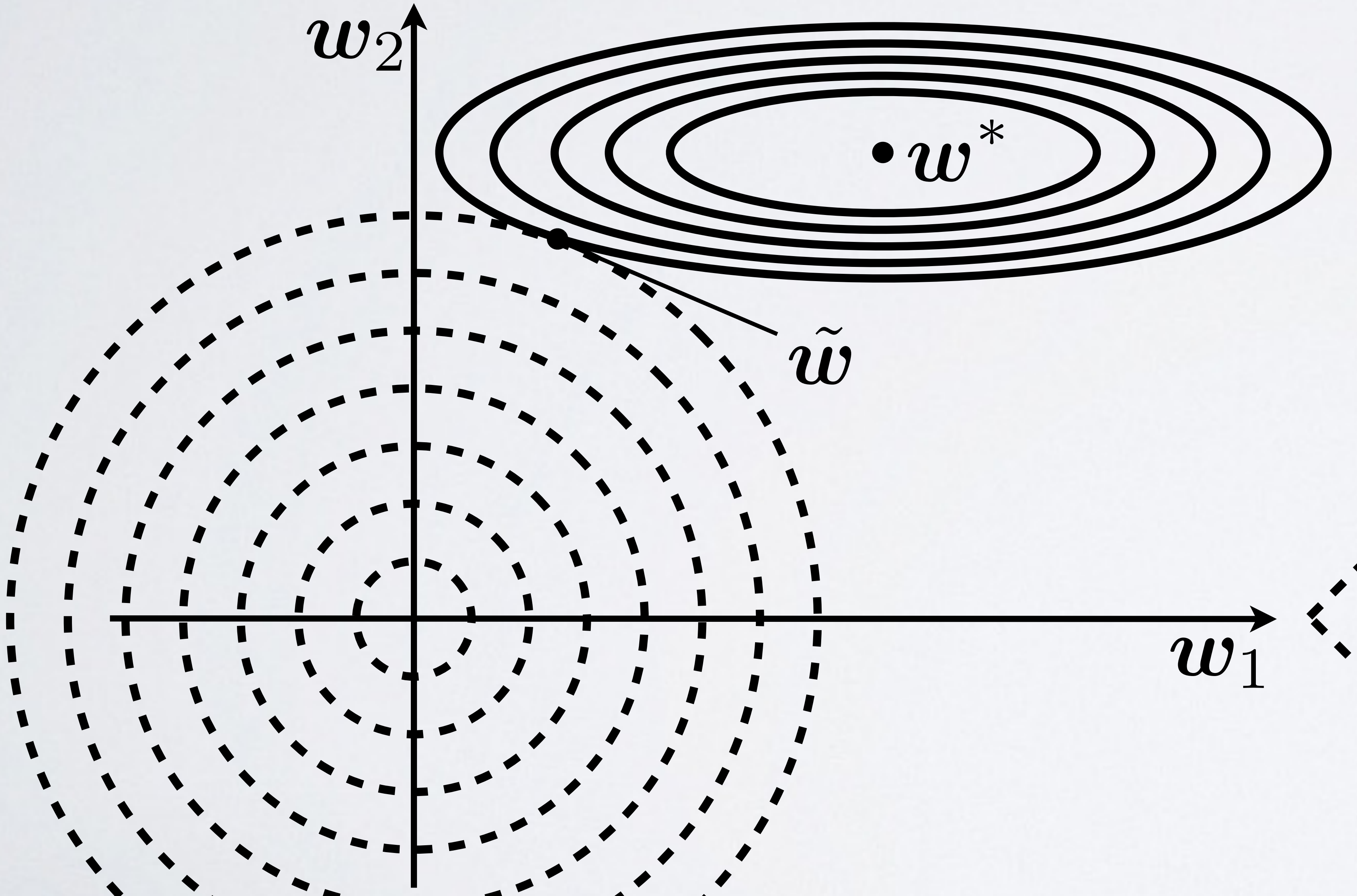
- Which admits an optimal solution (for each dimension) in the following form:

$$\boldsymbol{w}_i = \text{sign}(\boldsymbol{w}_i^*) \max(|\boldsymbol{w}_i^*| - \frac{\beta}{\gamma_i}, 0)$$

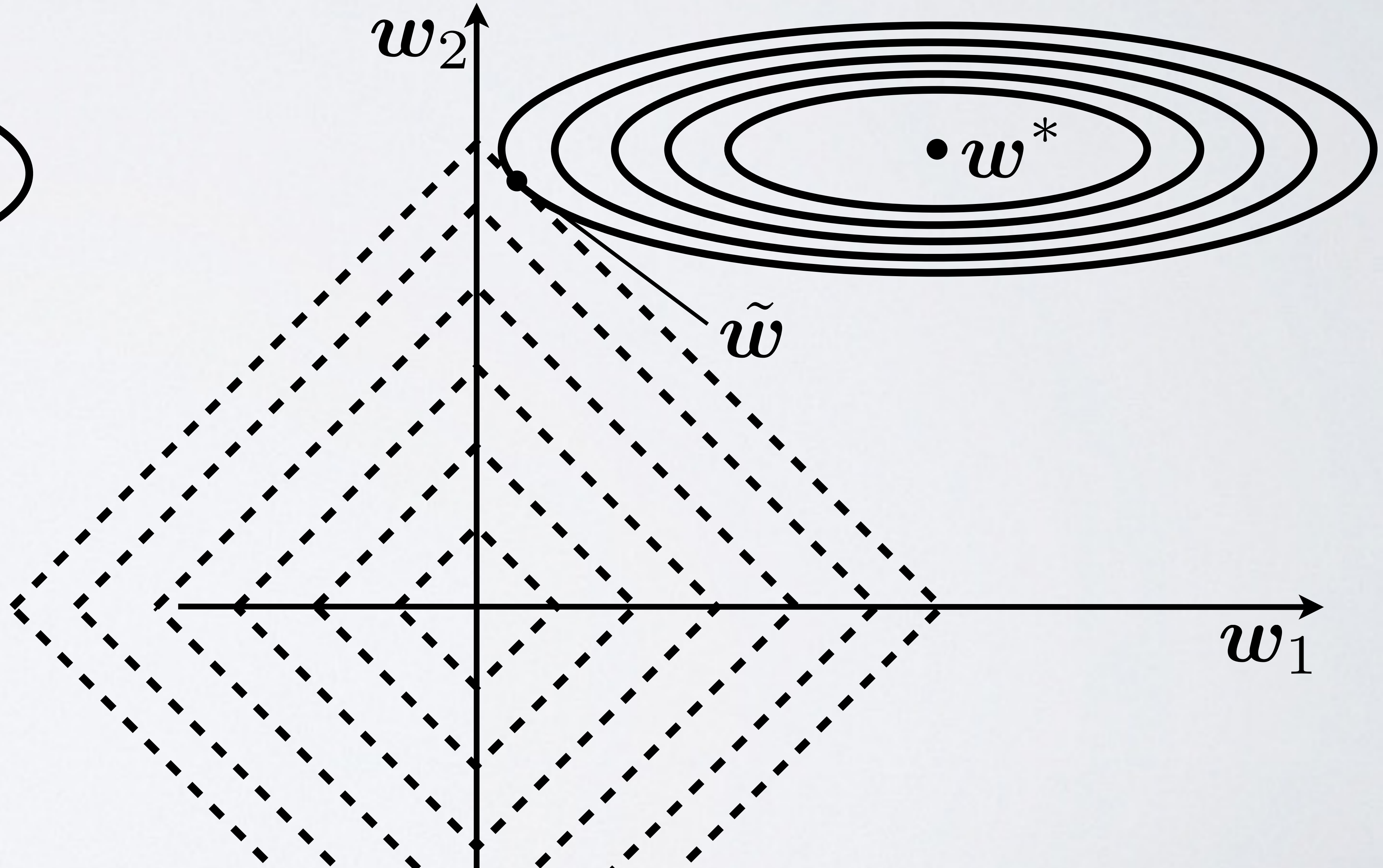
REGULARIZATION

Topics: L1 regularization $\Omega(\boldsymbol{\theta}) = ||\boldsymbol{w}||_1$

L2 regularization



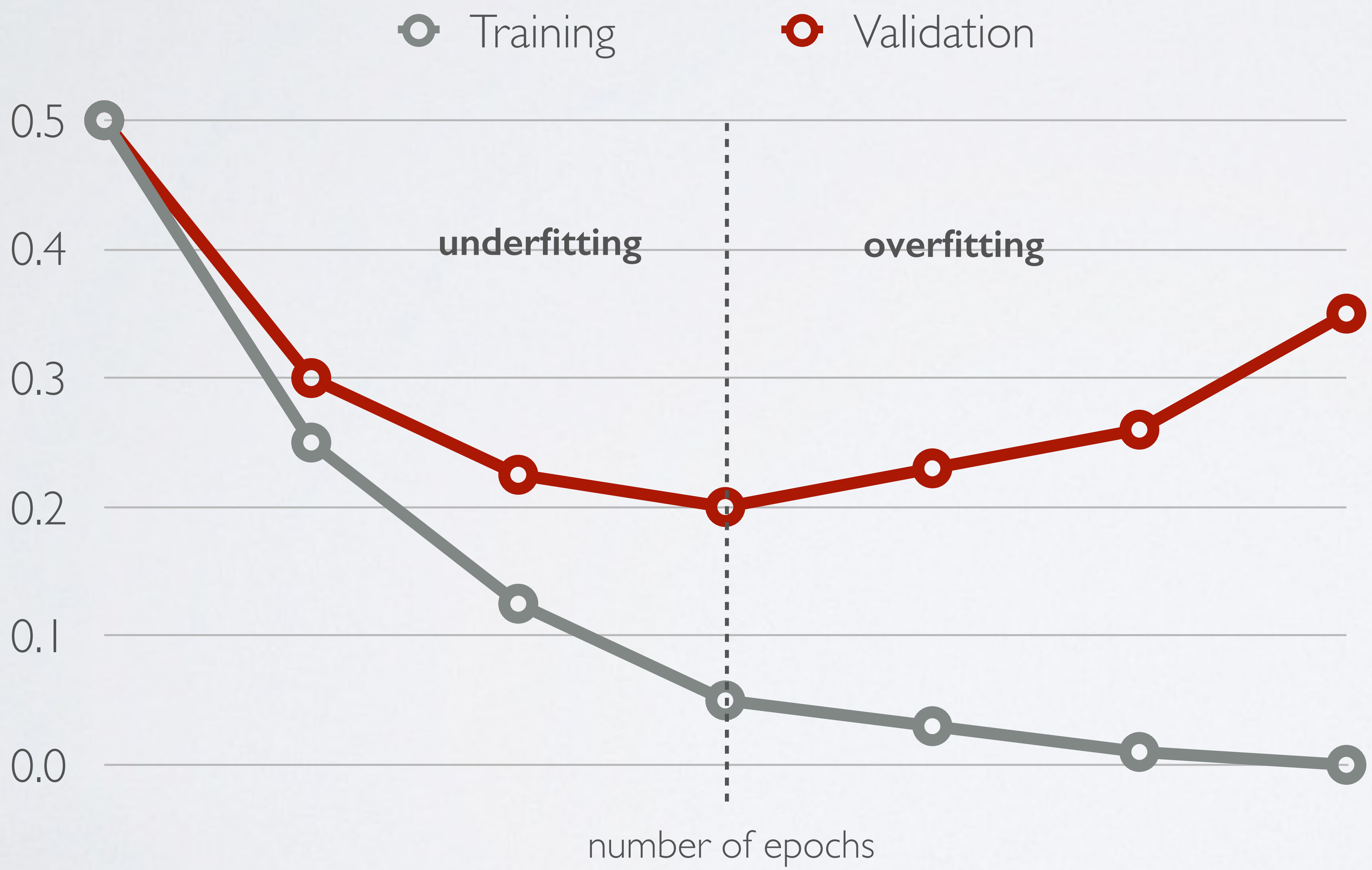
L1 regularization



KNOWING WHEN TO STOP

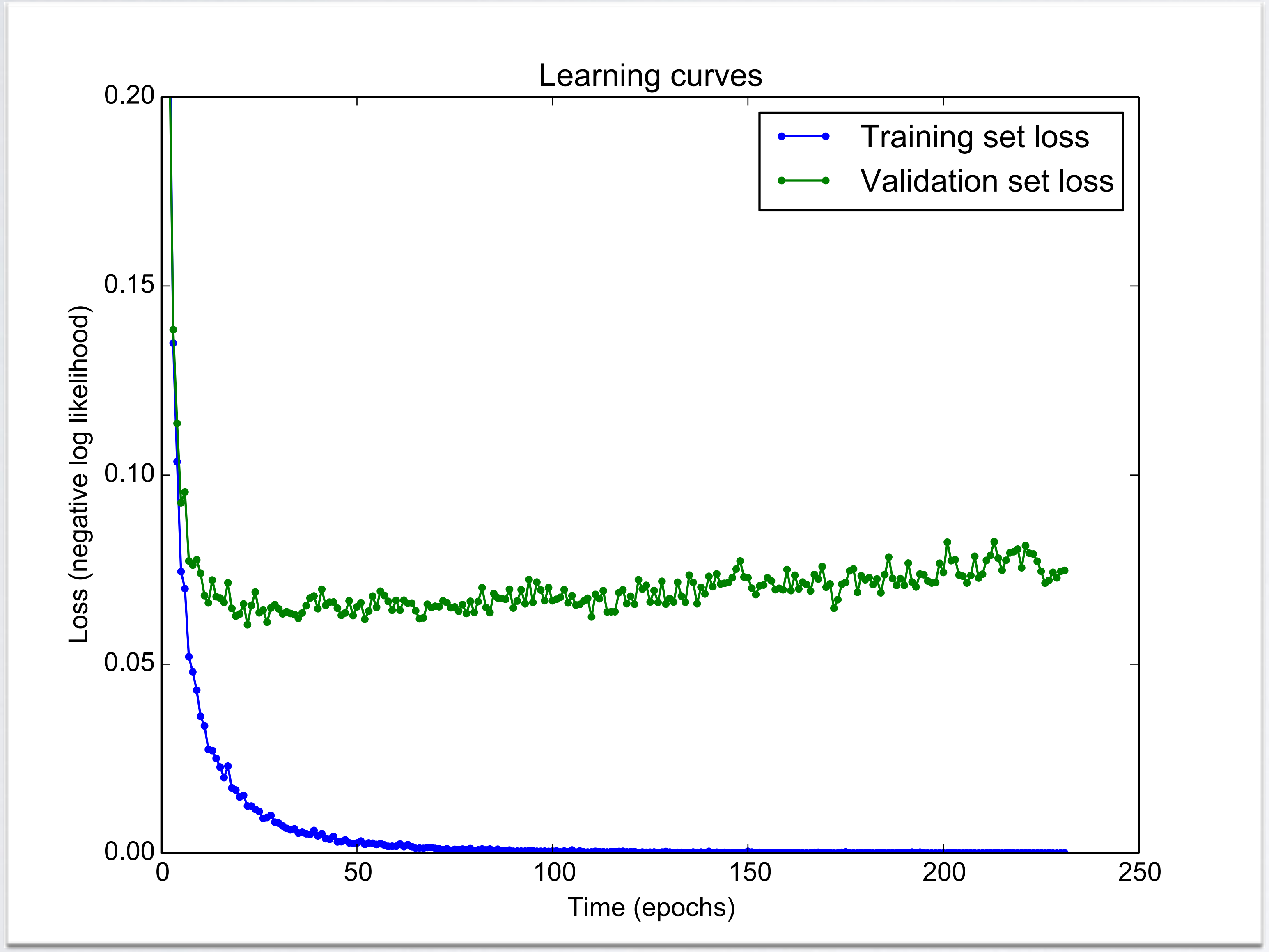
Topics: early stopping

- To select the number of epochs, stop training when validation set error increases (with some look ahead)



REGULARIZATION

Topics: Early stopping in practice



Algorithm 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^*

REGULARIZATION

Topics: Early stopping with retraining

- Sometimes you really don't want to "waste" the validation set by not training on it.
- There are two basic strategies for retraining with the validation data.
  1. Retrain with train+valid for the same number of (updates / epochs) as determined by initial early stopping.
  2. Continue training w/ train+valid until the loss on valid = early-stopped loss on train. Not guaranteed to stop.

Algorithm 1 A meta-algorithm for using early stopping to determine how long to train, then retraining on all the data.

Let $\boldsymbol{X}^{(\text{train})}$ and $\boldsymbol{y}^{(\text{train})}$ be the training set
Split $\boldsymbol{X}^{(\text{train})}$ and $\boldsymbol{y}^{(\text{train})}$ into $\boldsymbol{X}^{(\text{subtrain})}$, $\boldsymbol{y}^{(\text{subtrain})}$, $\boldsymbol{X}^{(\text{valid})}$, $\boldsymbol{y}^{(\text{valid})}$
Run early stopping starting from random $\boldsymbol{\theta}$ using $\boldsymbol{X}^{(\text{subtrain})}$ and $\boldsymbol{y}^{(\text{subtrain})}$ for training data and $\boldsymbol{X}^{(\text{valid})}$ and $\boldsymbol{y}^{(\text{valid})}$ for validation data. This returns i^*, the optimal number of steps.
Set $\boldsymbol{\theta}$ to random values again
Train on $\boldsymbol{X}^{(\text{train})}$ and $\boldsymbol{y}^{(\text{train})}$ for i^* steps.

Algorithm 2 A meta-algorithm for using early stopping to determining at what objective value we start to overfit, then continuing training.

Let $\boldsymbol{X}^{(\text{train})}$ and $\boldsymbol{y}^{(\text{train})}$ be the training set
Split $\boldsymbol{X}^{(\text{train})}$ and $\boldsymbol{y}^{(\text{train})}$ into $\boldsymbol{X}^{(\text{subtrain})}$, $\boldsymbol{y}^{(\text{subtrain})}$, $\boldsymbol{X}^{(\text{valid})}$, $\boldsymbol{y}^{(\text{valid})}$
Run early stopping (Alg. ??) starting from random $\boldsymbol{\theta}$ using $\boldsymbol{X}^{(\text{subtrain})}$ and $\boldsymbol{y}^{(\text{subtrain})}$ for training data and $\boldsymbol{X}^{(\text{valid})}$ and $\boldsymbol{y}^{(\text{valid})}$ for validation data. This updates $\boldsymbol{\theta}$
$\epsilon \leftarrow J(\boldsymbol{\theta}, \boldsymbol{X}^{(\text{subtrain})}, \boldsymbol{y}^{(\text{subtrain})})$
while $J(\boldsymbol{\theta}, \boldsymbol{X}^{(\text{valid})}, \boldsymbol{y}^{(\text{valid})}) > \epsilon$ **do**
  Train on $\boldsymbol{X}^{(\text{train})}$ and $\boldsymbol{y}^{(\text{train})}$ for n steps.
end while

Warning: these methods are dangerous!

REGULARIZATION

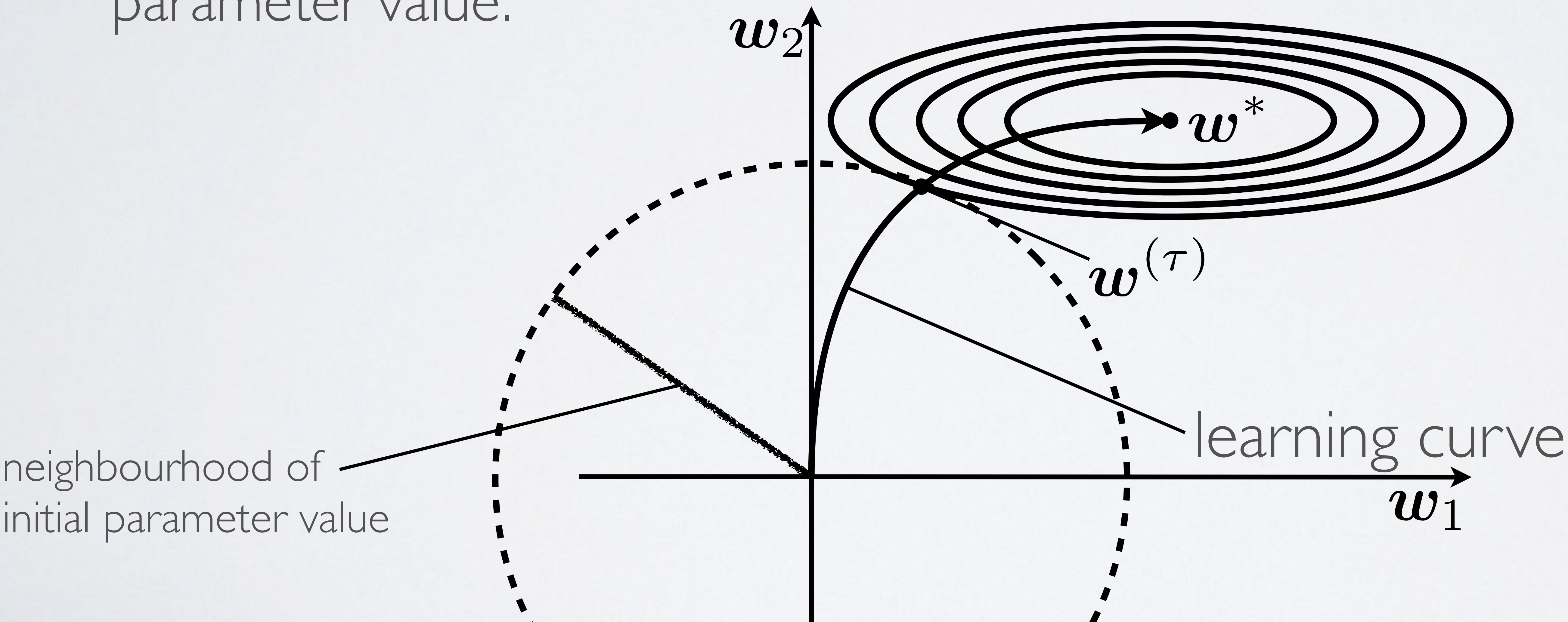
Topics: Early stopping with surrogate loss

- A useful property: can help to mitigate a mismatch between the surrogate loss and the underlying performance measure that we actually care about.
  - Example, 0-1 classification loss (derivative of zero almost everywhere). We therefore train with surrogates such as the log likelihood of correct class label.
  - However, 0-1 loss is inexpensive to compute, so it can easily be used as an early stopping criterion.
  - Often the 0-1 loss decreases long after the log likelihood has begun to worsen on the validation set.

REGULARIZATION

Topics: How early stopping acts as a regularizer.

- What is the actual mechanism by which early stopping regularizes the model?
  - Early stopping has the effect of restricting the optimization procedure to a relatively small volume of parameter space in the neighbourhood of the initial parameter value.

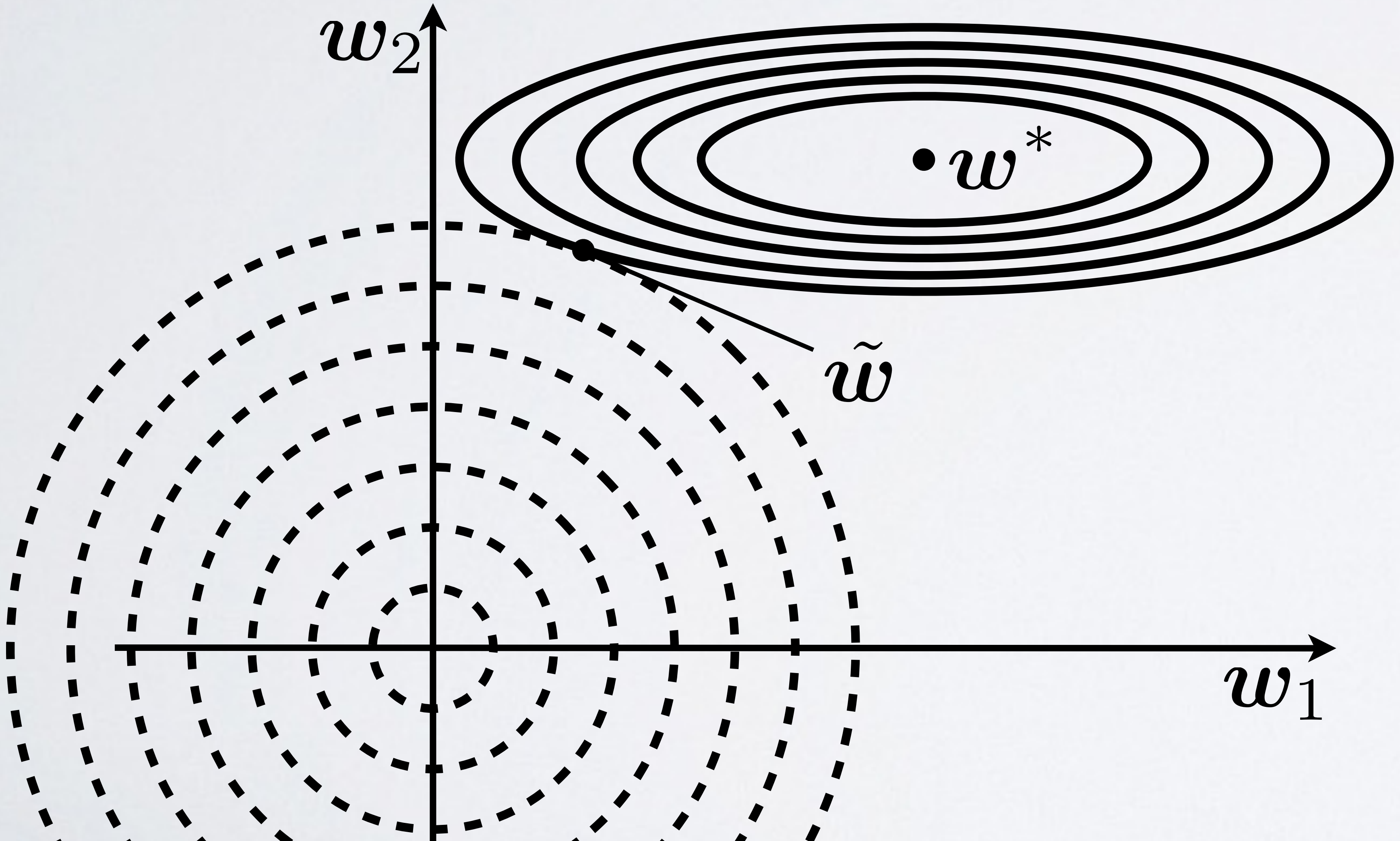


REGULARIZATION

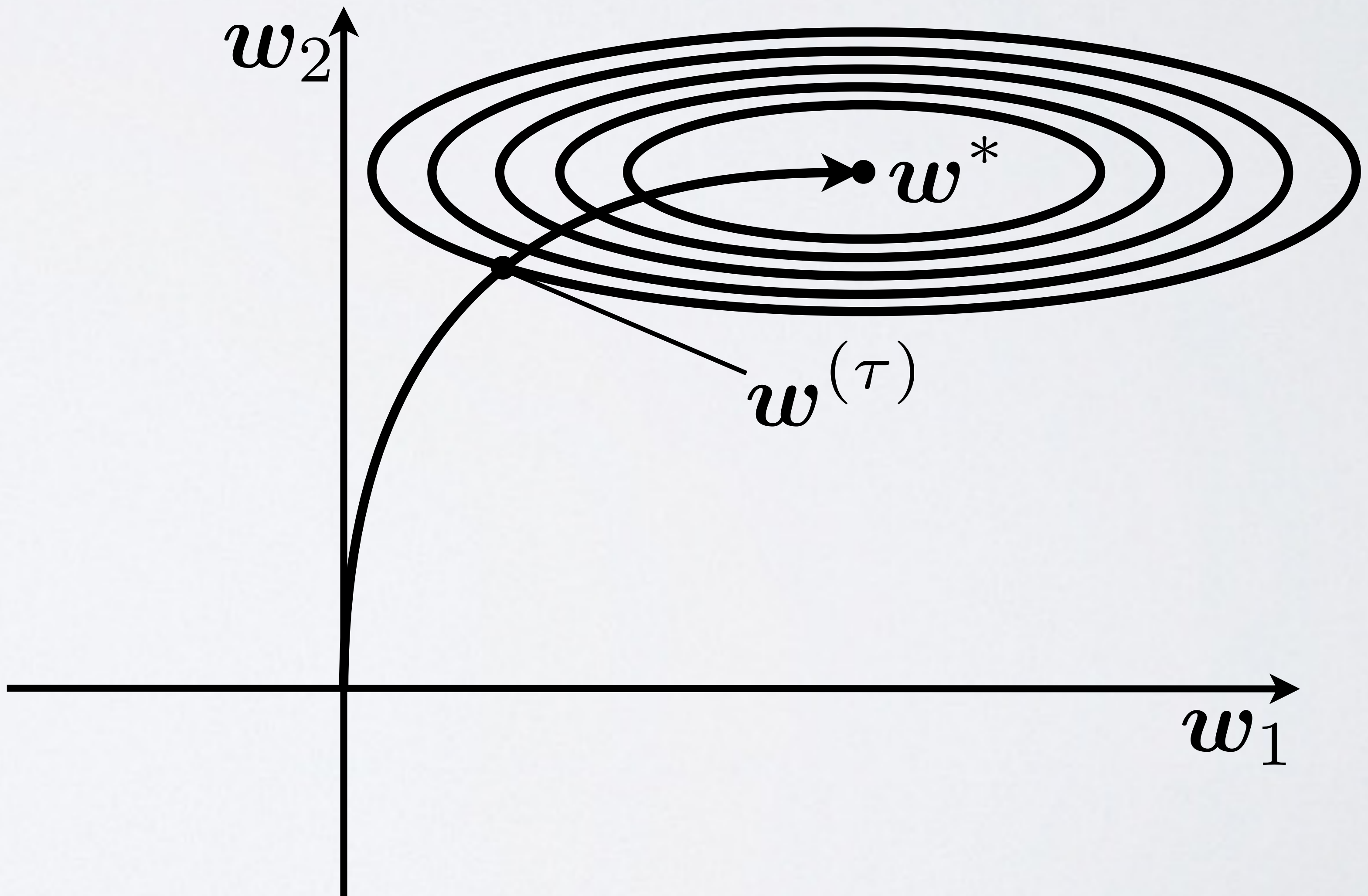
Topics: How early stopping acts as a regularizer.

- Assuming a simple linear model with a quadratic error function and simple gradient descent -- early stopping is equivalent to L2 regularization.

L2 regularization



Early Stopping



REGULARIZATION

Topics: Early stopping equivalence to L2 regularization, **mathematical details**.

- Consider a quadratic approximation to the loss function in the neighbourhood of the empirically optimal value of the weights $\boldsymbol{w}^*$

$$\hat{J}(\boldsymbol{\theta}) = J(\boldsymbol{w}^*) + \frac{1}{2}(\boldsymbol{w} - \boldsymbol{w}^*)^\top \boldsymbol{H}(\boldsymbol{w} - \boldsymbol{w}^*)$$

Taking gradient

$$\nabla_{\boldsymbol{w}} \hat{J}(\boldsymbol{w}) = \boldsymbol{H}(\boldsymbol{w} - \boldsymbol{w}^*)$$

REGULARIZATION

Topics: Early stopping equivalence to L2 regularization, **mathematical details**.

$$\nabla_{\boldsymbol{w}} \hat{J}(\boldsymbol{w}) = \boldsymbol{H}(\boldsymbol{w} - \boldsymbol{w}^*)$$

- Let us consider initial parameter vector chosen at the origin,
- We will consider updating the parameters via gradient descent:

$$\begin{aligned}
\boldsymbol{w}^{(\tau)} &= \boldsymbol{w}^{(\tau-1)} - \eta \nabla_{\boldsymbol{w}} J(\boldsymbol{w}^{(\tau-1)}) \\
&= \boldsymbol{w}^{(\tau-1)} - \eta \boldsymbol{H}(\boldsymbol{w}^{(\tau-1)} - \boldsymbol{w}^*)
\end{aligned}$$

$$\boldsymbol{w}^{(\tau)} - \boldsymbol{w}^* = (\boldsymbol{I} - \eta \boldsymbol{H})(\boldsymbol{w}^{(\tau-1)} - \boldsymbol{w}^*)$$

REGULARIZATION

Topics: Early stopping equivalence to L2 regularization, **mathematical details**.

$$\boldsymbol{w}^{(\tau)} - \boldsymbol{w}^* = (\boldsymbol{I} - \eta \boldsymbol{H})(\boldsymbol{w}^{(\tau-1)} - \boldsymbol{w}^*)$$

- $\boldsymbol{H}$ is real and symmetric, so we can decompose it into a diagonal matrix $\boldsymbol{\Lambda}$ and an orthogonal basis of eigenvectors, $\boldsymbol{Q}$, such that: $\boldsymbol{H} = \boldsymbol{Q}\boldsymbol{\Lambda}\boldsymbol{Q}^\top$

$$\boldsymbol{w}^{(\tau)} - \boldsymbol{w}^* = (\boldsymbol{I} - \eta \boldsymbol{Q}\boldsymbol{\Lambda}\boldsymbol{Q}^\top)(\boldsymbol{w}^{(\tau-1)} - \boldsymbol{w}^*)$$

$$\boldsymbol{Q}^\top(\boldsymbol{w}^{(\tau)} - \boldsymbol{w}^*) = (\boldsymbol{I} - \eta \boldsymbol{\Lambda})\boldsymbol{Q}^\top(\boldsymbol{w}^{(\tau-1)} - \boldsymbol{w}^*)$$

- Assuming that $|1 - \eta\lambda_i| < 1$ and that $\boldsymbol{w}^{(0)} = \boldsymbol{0}$. After τ steps:

$$\boldsymbol{Q}^\top \boldsymbol{w}^{(\tau)} = [\boldsymbol{I} - (\boldsymbol{I} - \eta\boldsymbol{\Lambda})^\tau]\boldsymbol{Q}^\top \boldsymbol{w}^*$$

REGULARIZATION

Topics: Early stopping equivalence to L2 regularization, **mathematical details**.

$$\boldsymbol{Q}^\top \boldsymbol{w}^{(\tau)} = [\boldsymbol{I} - (\boldsymbol{I} - \eta\boldsymbol{\Lambda})^\tau]\boldsymbol{Q}^\top \boldsymbol{w}^*$$

- Recall the L2 regularized solution was: $\tilde{\boldsymbol{w}} = \boldsymbol{Q}(\boldsymbol{\Lambda} + \alpha\boldsymbol{I})^{-1}\boldsymbol{\Lambda}\boldsymbol{Q}^\top \boldsymbol{w}^*$

$$\boldsymbol{Q}^\top \tilde{\boldsymbol{w}} = (\boldsymbol{\Lambda} + \alpha\boldsymbol{I})^{-1}\boldsymbol{\Lambda}\boldsymbol{Q}^\top \boldsymbol{w}^*$$

$$\boldsymbol{Q}^\top \tilde{\boldsymbol{w}} = [\boldsymbol{I} - (\boldsymbol{\Lambda} + \alpha\boldsymbol{I})^{-1}\alpha]\boldsymbol{Q}^\top \boldsymbol{w}^*$$

- These are equivalent when $(\boldsymbol{I} - \eta\boldsymbol{\Lambda})^\tau = (\boldsymbol{\Lambda} + \alpha\boldsymbol{I})^{-1}\alpha$

$$\tau \log(\boldsymbol{I} - \eta\boldsymbol{\Lambda}) = -\log(\boldsymbol{I} + \boldsymbol{\Lambda}/\alpha)$$

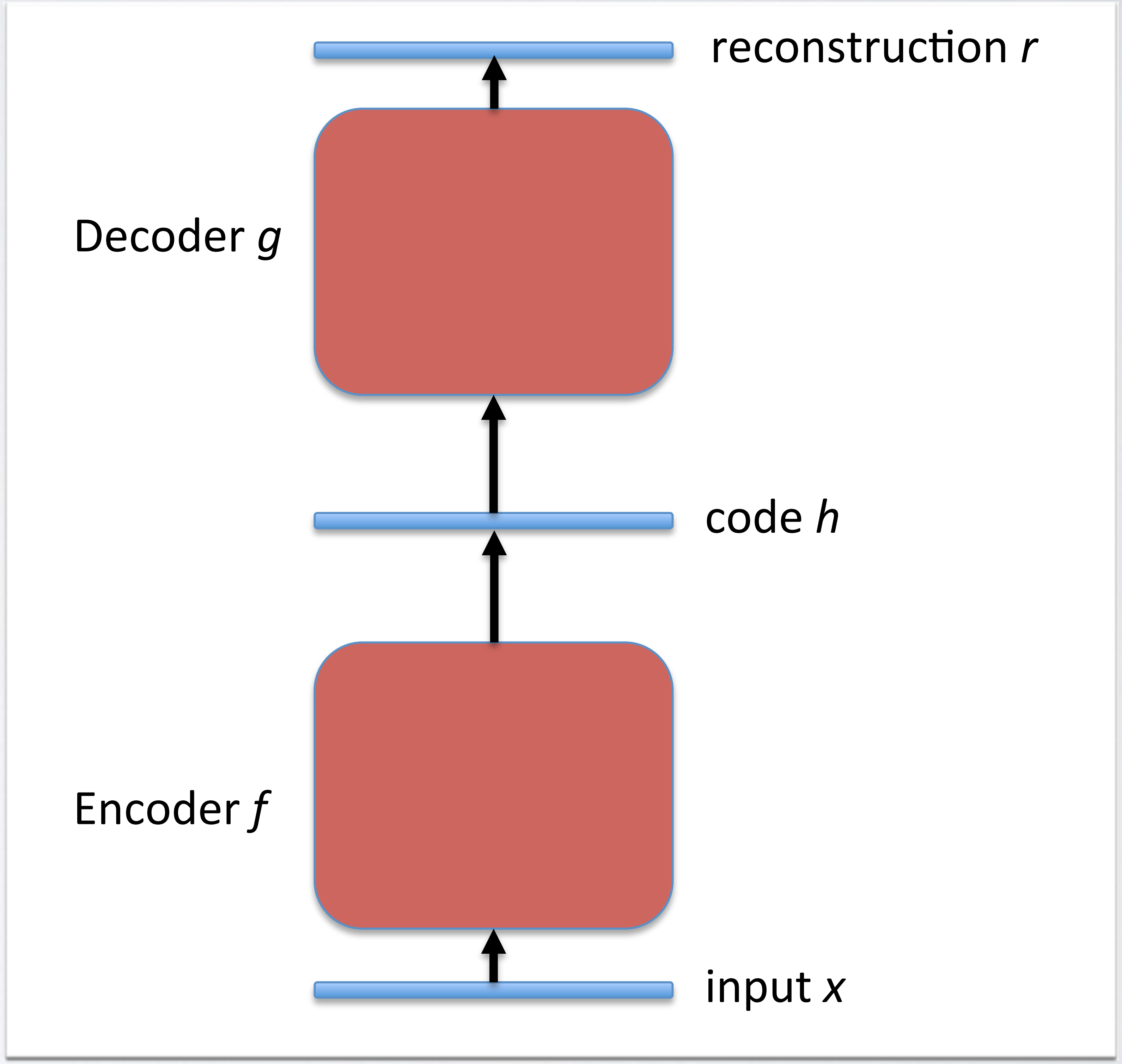
(by Taylor series expansion) $\tau \approx 1/\eta\alpha \quad \text{for small } \lambda_i \ \forall i$

Unsupervised learning as a regularization strategy

REGULARIZATION

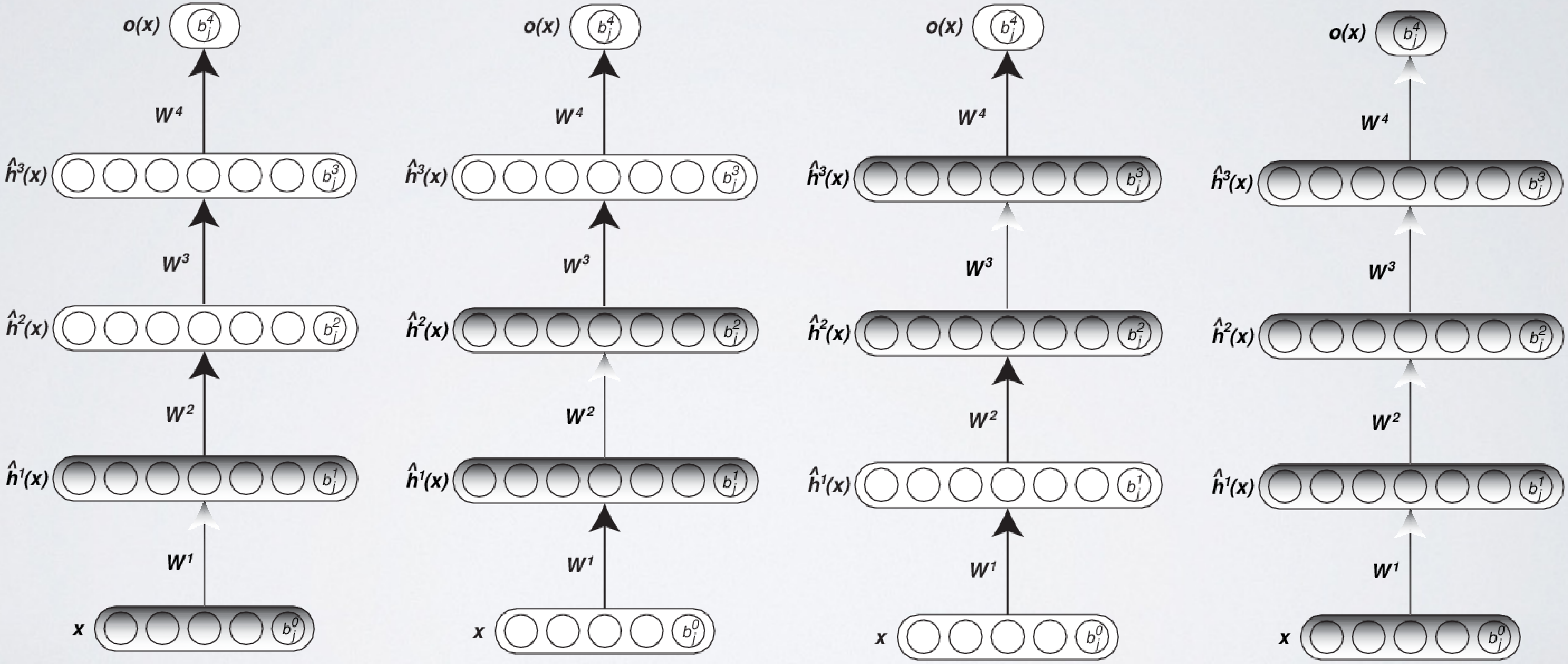
Topics: Unsupervised pretraining.

- Idea: pretrain your discriminative model parameters as an autoencoder.
- Autoencoders are featured prominently in the deep learning literature
- Goal: learn an encoder (f) and decoder (g) to minimize reconstruction error.
- Often, an additional penalty term is used to give the code (h) desirable characteristics (we will see this later in the course)



REGULARIZATION

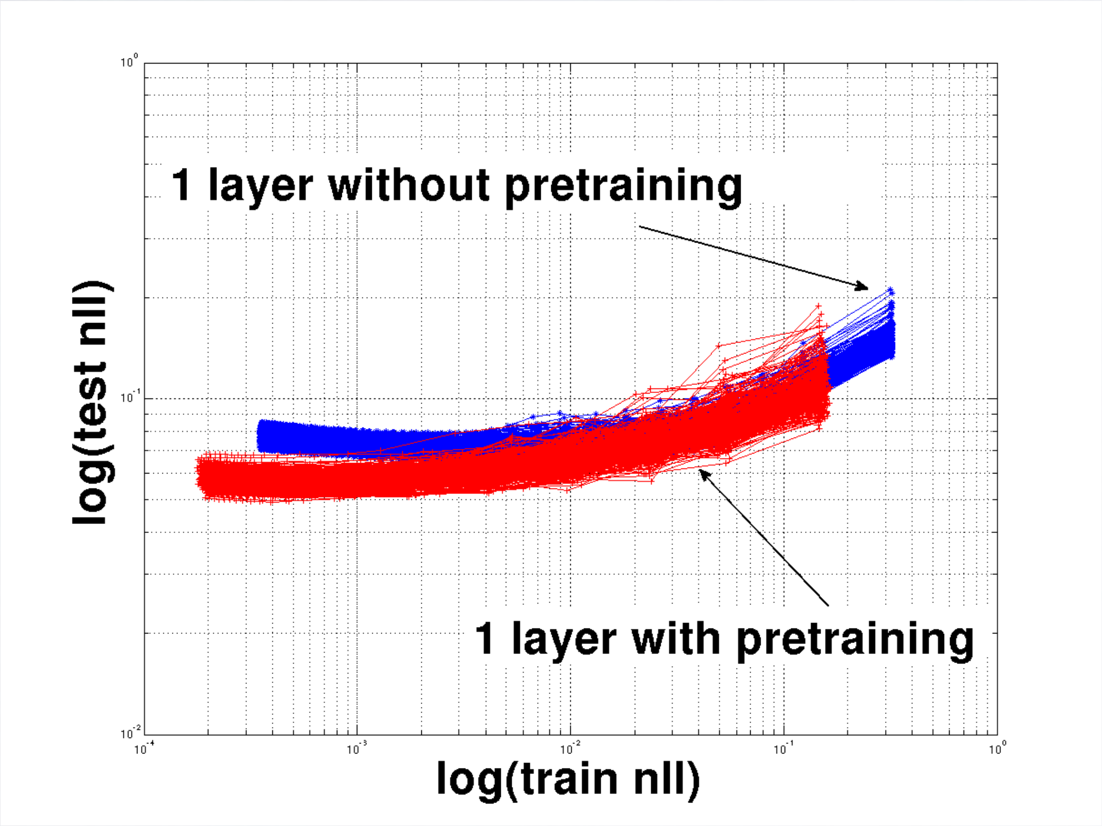
Topics: Greedy layer-wise unsupervised pretraining.

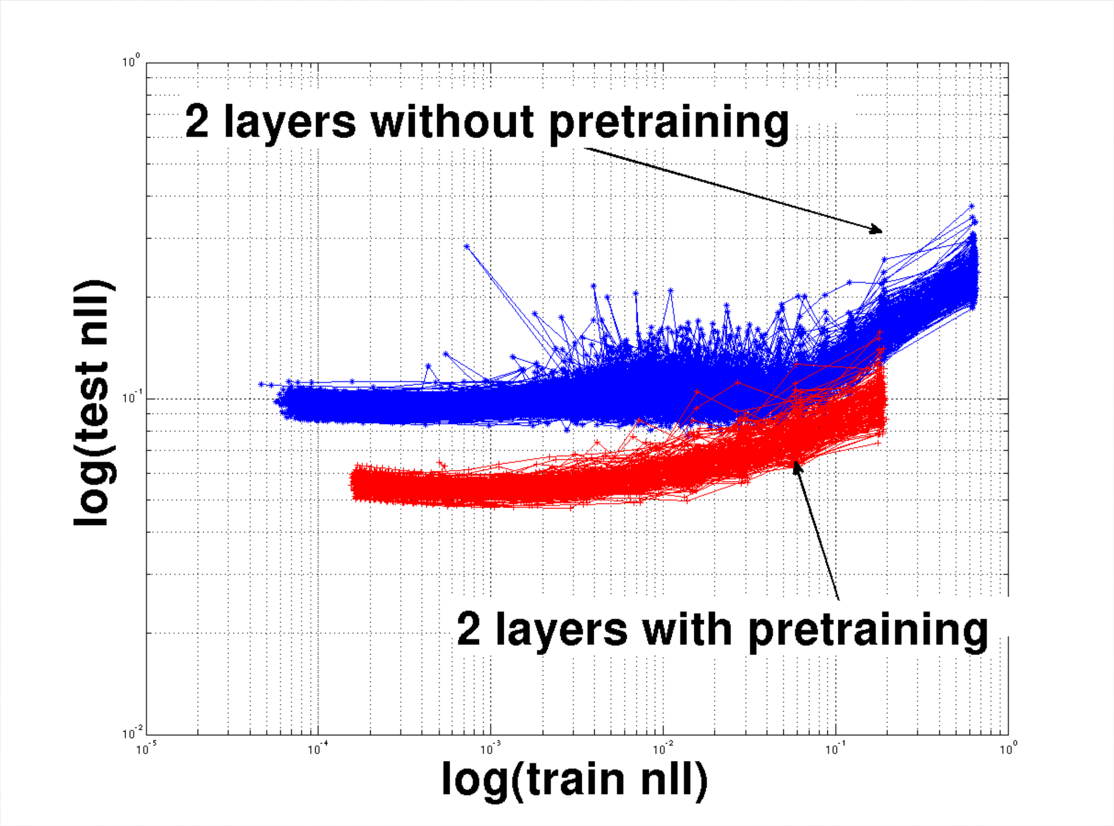


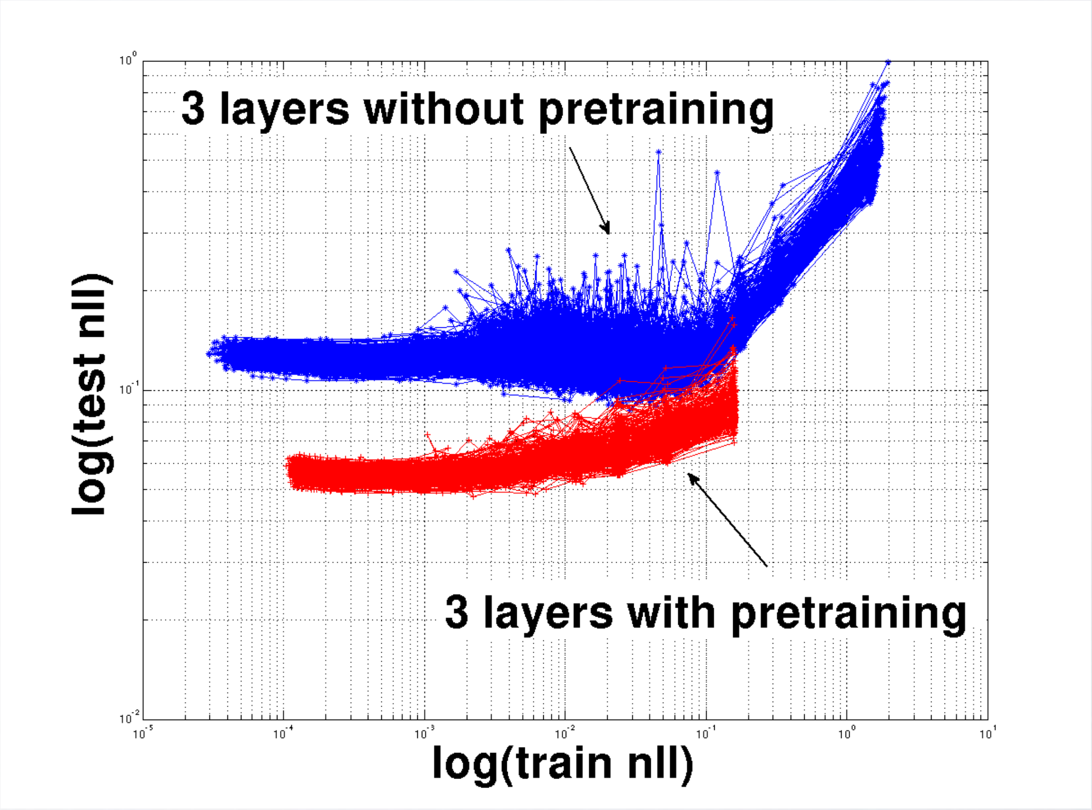
REGULARIZATION

Topics: Greedy layer-wise unsupervised pretraining as a regularization strategy:

- Training error / Test error profile matches that of a regularizer (Erhan et al. 2009).



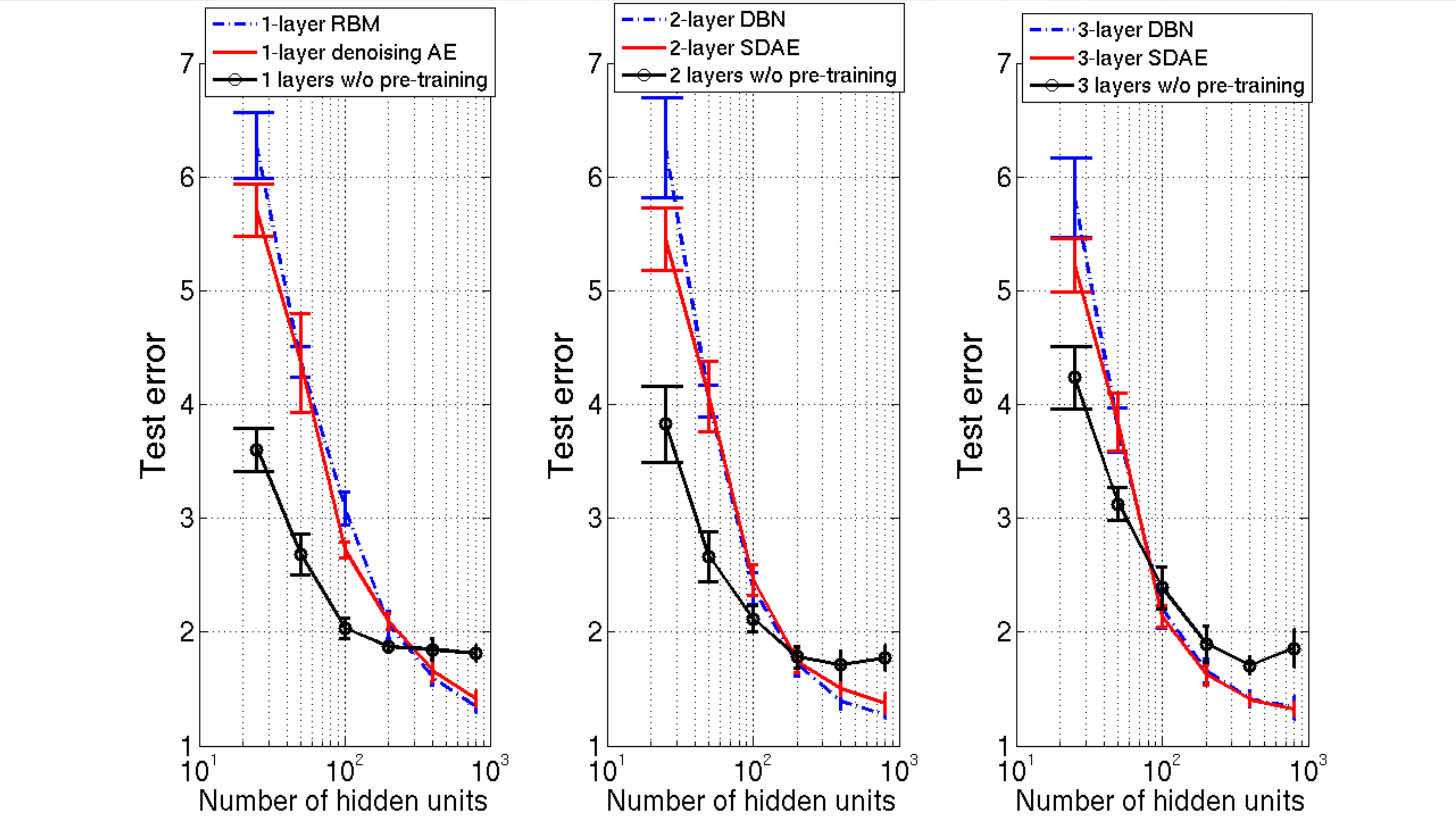




REGULARIZATION

Topics: Greedy layer-wise unsupervised pretraining as a regularization strategy:

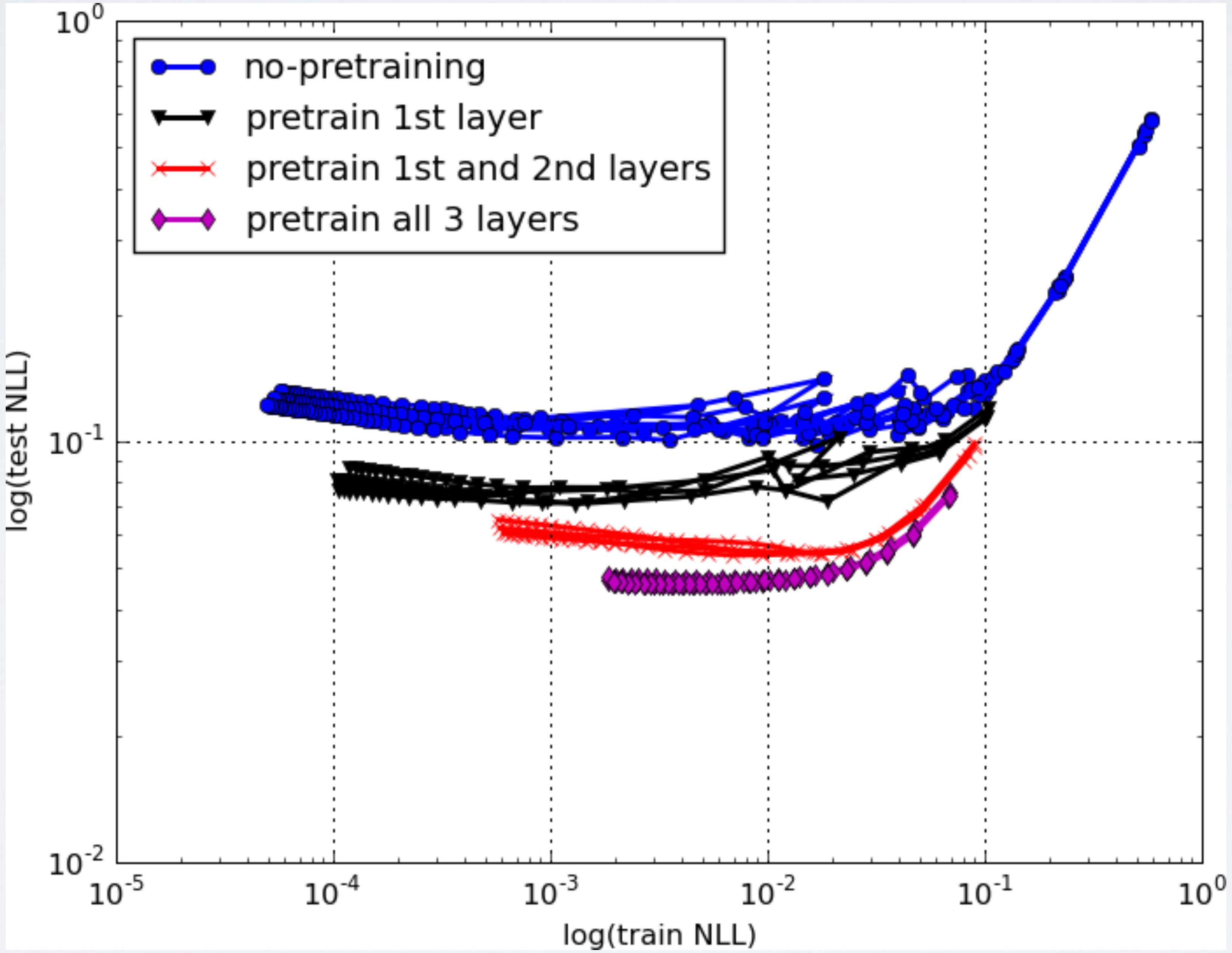
- Training error / Test error profile matches that of a regularizer (Erhan et al. 2009).



REGULARIZATION

Topics: Greedy layer-wise unsupervised pretraining as a regularization strategy:

- Training error / Test error profile matches that of a regularizer (Erhan et al. 2009).



REGULARIZATION

Topics: Multi-task learning / unsupervised learning.

- Same principle that applied to unsupervised learning applies to multi-task learning and transfer learning.
- Both are strategies to leverage other related tasks to **regularize** the parameters of the target task
- True even when there are multiple target tasks as in multi-task learning.
  - Each task regularizers the others.