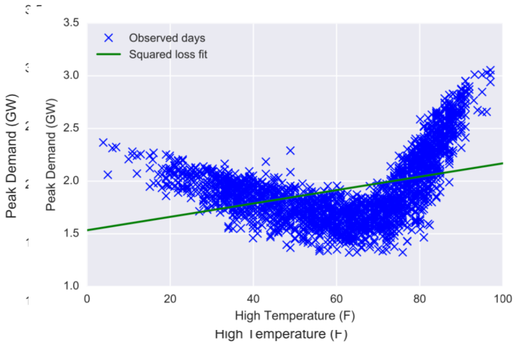
BLG 561E
Deep Learning
Fall 2021

Lecture: Deep Learning Week 2

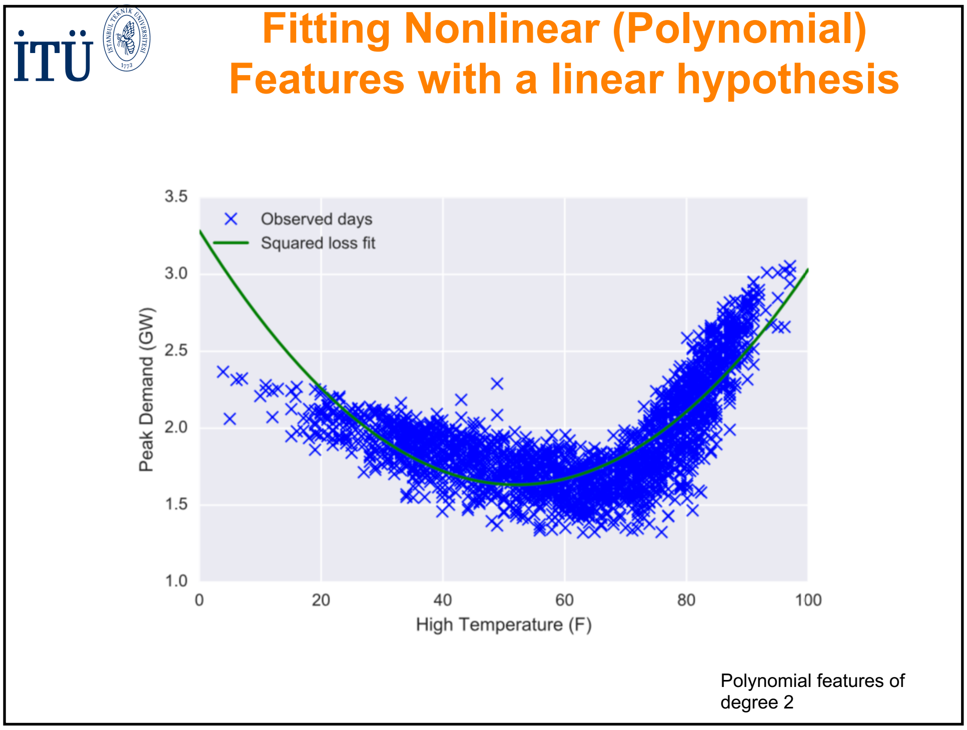
CRN 14899

Gözde Ünal

Fitting Nonlinear (Polynomial) Features with a linear hypothesis



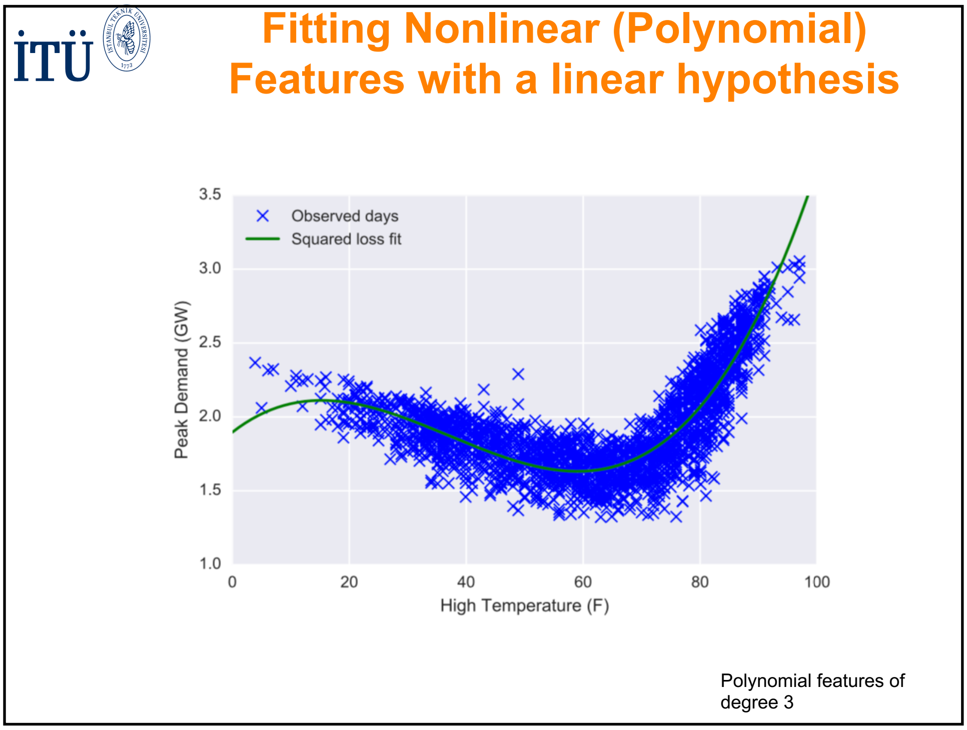
Linear Regression fit



Fitting Nonlinear (Polynomial) Features with a linear hypothesis

Polynomial features of degree 2

3

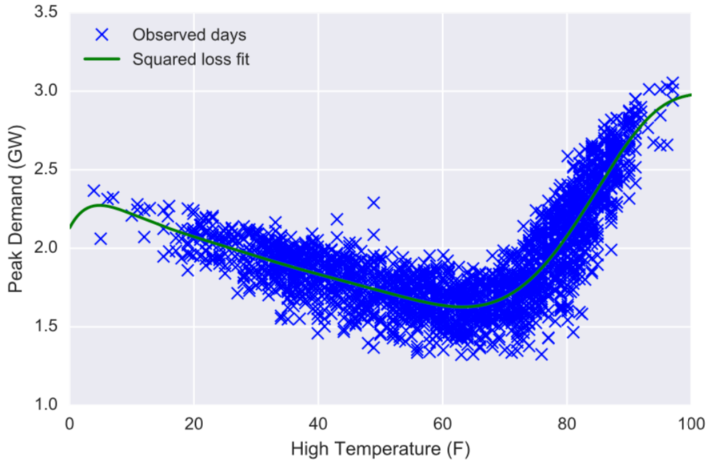


Fitting Nonlinear (Polynomial) Features with a linear hypothesis

Polynomial features of degree 3

4

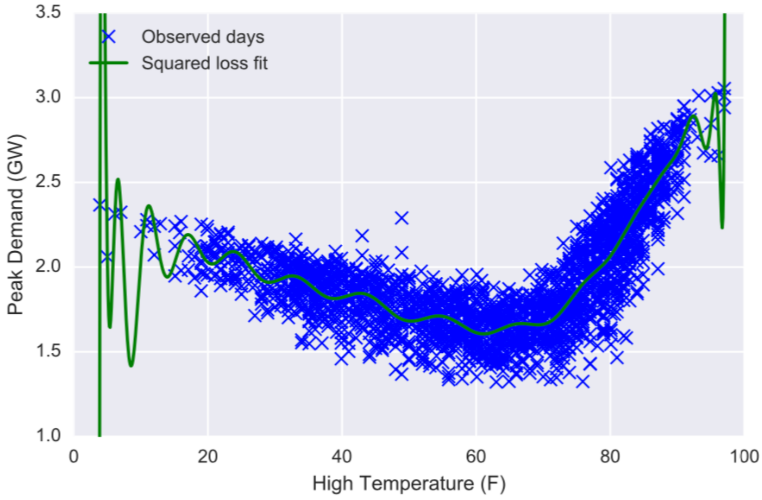
Fitting Nonlinear (Polynomial) Features with a linear hypothesis



Polynomial features of degree 10

5

Fitting Nonlinear (Polynomial) Features with a linear hypothesis



Polynomial features of degree 50

6

Generalization

→ Representation Problem!

The higher degree polynomials exhibited overfitting: although they have very low loss on the training data.

They led to hypothesis functions that did not generalize well

We trained our system by minimizing the objective:

$$\underset{\theta}{\text{minimize}} \sum_{i=1}^{m} \ell(h_\theta(x^{(i)}), y^{(i)})$$

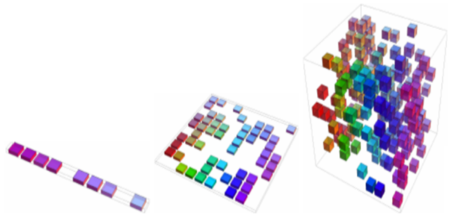
Q: Is this error what we really care about in developing robust ML algorithms?

No. We care about how well our model will generalize to new examples that we did not use to train the system.

Note that we assume that they are also drawn from the "same distribution" as the examples we used for training.

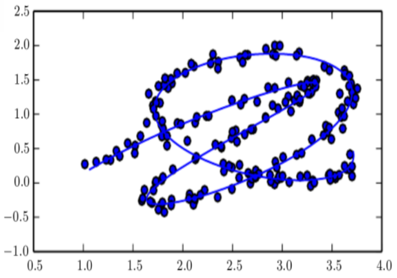
7

Motivation for Deep Learning –



Curse of Dimensionality

The purpose of ML is to generalize based on training data, but as the dimension grows, the number of configurations of dimensions is exponential. Our training data will not likely to cover all those variations in high dimensional spaces.



Manifold learning

8

Generalization

A central challenge in machine learning is to perform well on new, **previously unseen** inputs.

- This property is known as generalization. The error on the test data is referred to as *generalization error*

But how can we say something about the test data, if all we see is training data?

- If these datasets were generated arbitarily, we cant...
- However, if they come from the same distribution $p(\boldsymbol{x}, y)$, then we can say something! For instance, their mean should be equal

9

Generalization

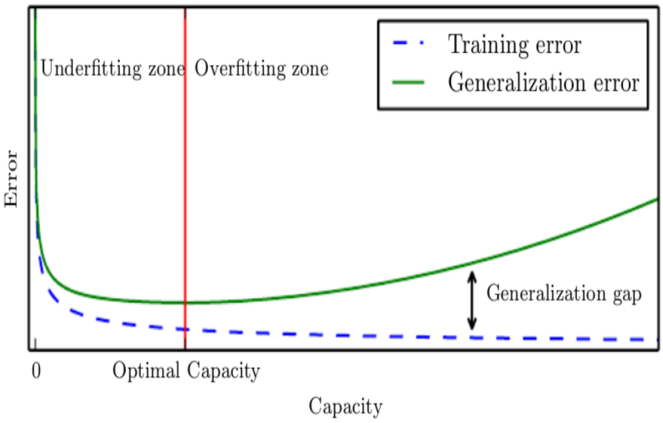
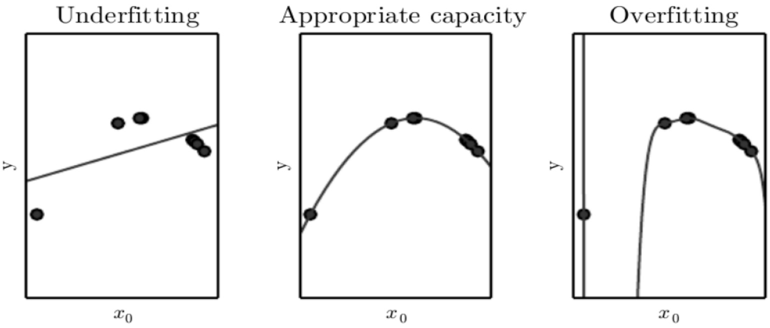


Figure: Goodfellow,Deep Learning Book Chapter 5

10

Generalization



11

Regularization

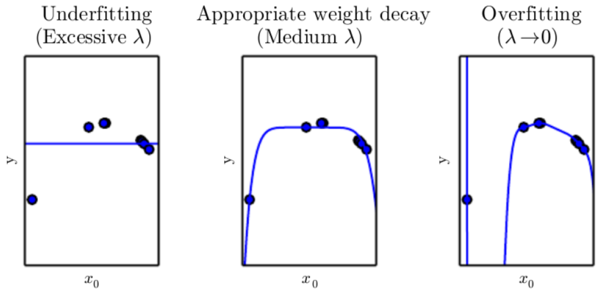
Regularization is any modification we make to the learning algorithm to reduce generalization error.

- The most common form of regularization is to modify the loss function so that larger parameters are penalized

$$J(\boldsymbol{w}) = \mathrm{MSE}_{\mathrm{train}}(\boldsymbol{w}) + \lambda \boldsymbol{w}^\top \boldsymbol{w}$$

12

Regularization



- There are other forms of regularization, for instance 'dropout' (randomly dropping some of the weights in your model) is very popular in deep learning

13

Hyperparameters

Hyperparameters: Choices about the algorithm that we typically set rather than learn

- Hyperparameters are the parameters of our model that controls the capacity, as well as the behaviour of the algorithm
- Examples: degree of a polynomial, number of layers or width in a neural net.

- These could be usually set by domain experts
- Can we "learn" those parameters as well?
- Yes! We can try optimizing them: this is called. HYPERPARAMETER OPTIMIZATION

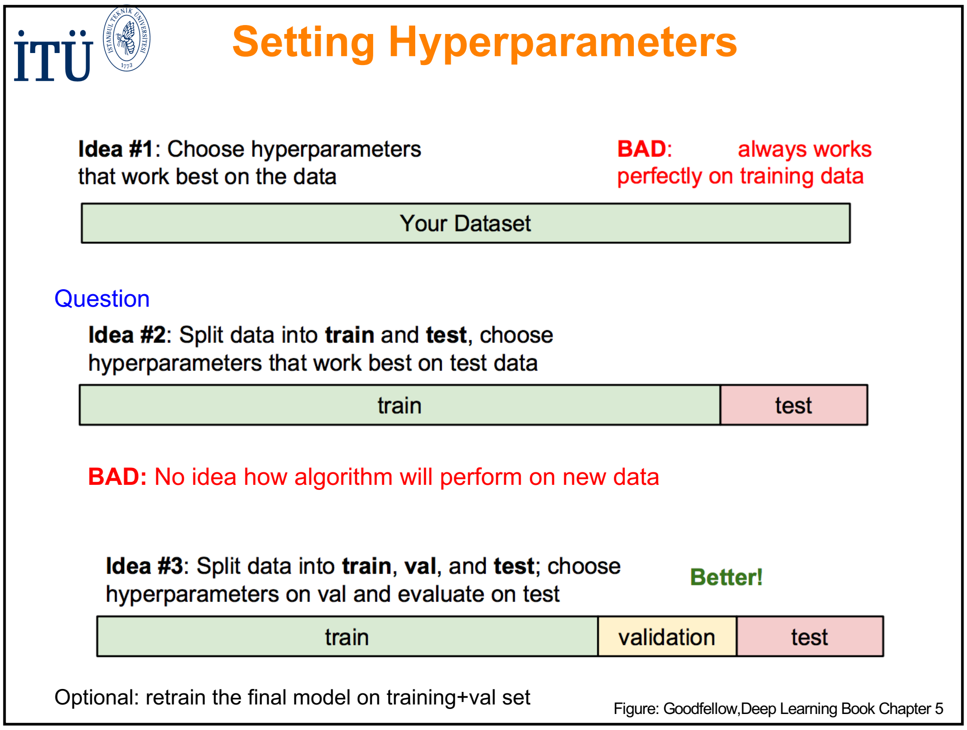
Q: Possibly the 2 most important hyperparameters to worry about in NNs ?

A: Learning rate and Regularizer Weight

Others: Batchsize, dropout rate, etc

Very problem-dependent.
Must try them all out and see what works best.

14



Setting Hyperparameters

Idea #1: Choose hyperparameters that work best on the data

BAD: always works perfectly on training data

Your Dataset

Question

Idea #2: Split data into **train** and **test**, choose hyperparameters that work best on test data

train | test

BAD: No idea how algorithm will perform on new data

Idea #3: Split data into **train**, **val**, and **test**; choose hyperparameters on val and evaluate on test

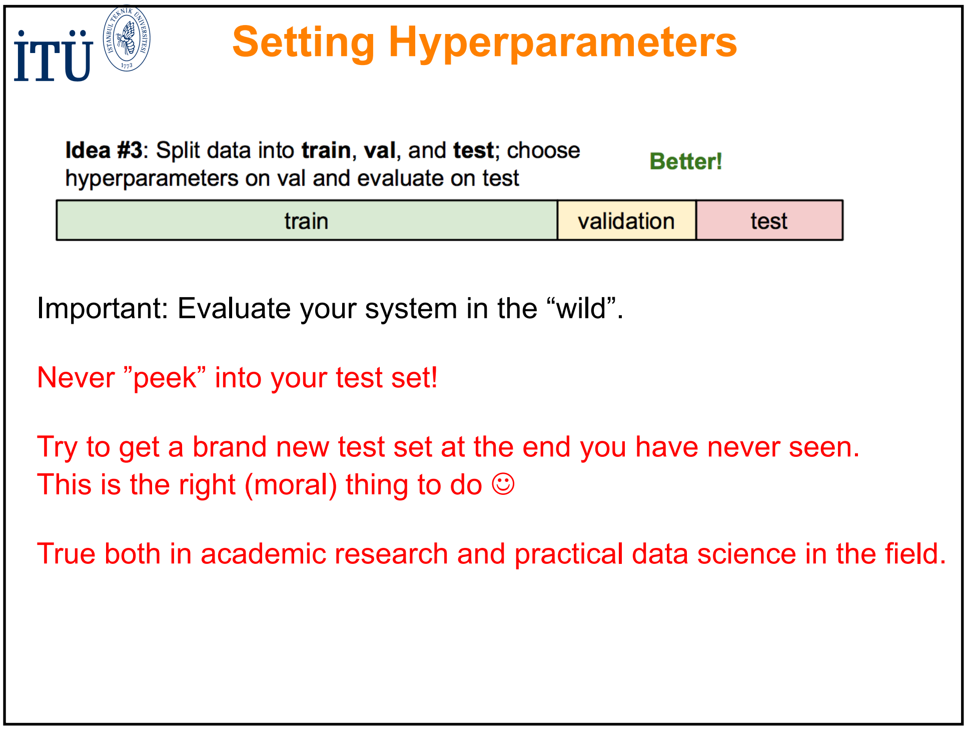
Better!

train | validation | test

Optional: retrain the final model on training+val set

Figure: Goodfellow,Deep Learning Book Chapter 5

15



Setting Hyperparameters

Idea #3: Split data into **train**, **val**, and **test**; choose hyperparameters on val and evaluate on test

Better!

train | validation | test

Important: Evaluate your system in the “wild”.

Never ”peek” into your test set!

Try to get a brand new test set at the end you have never seen.
This is the right (moral) thing to do ☺

True both in academic research and practical data science in the field.

16

Setting Hyperparameters

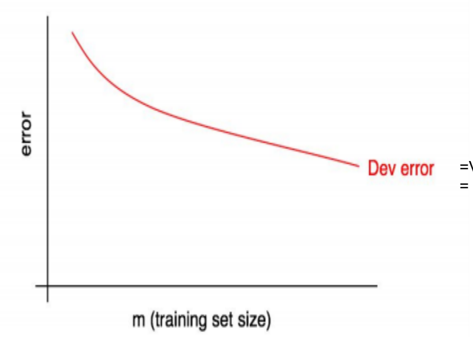
Your Dataset

Idea #4: **Cross-Validation**: Split data into **folds**, try each fold as validation and average the results

fold 1	fold 2	fold 3	fold 4	fold 5	test
fold 1	fold 2	fold 3	fold 4	fold 5	test
fold 1	fold 2	fold 3	fold 4	fold 5	test

Useful for small datasets, but not used too frequently in deep learning

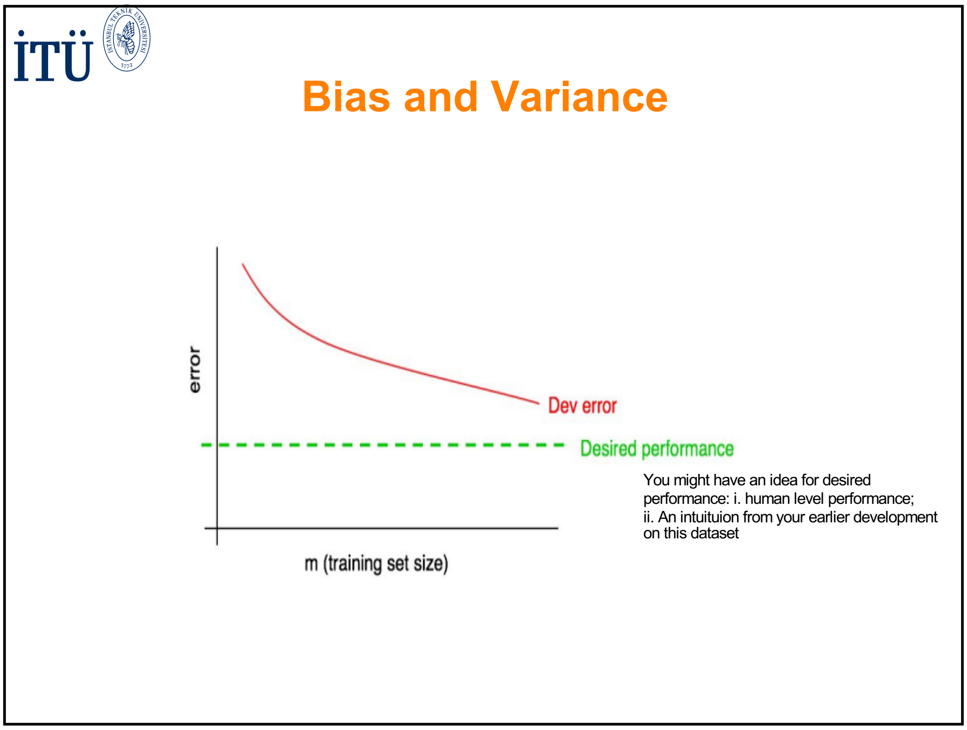
Bias and Variance



=Validation set error
= Holdout set error

You expect Valid error to decrease as you add more training samples to the data.

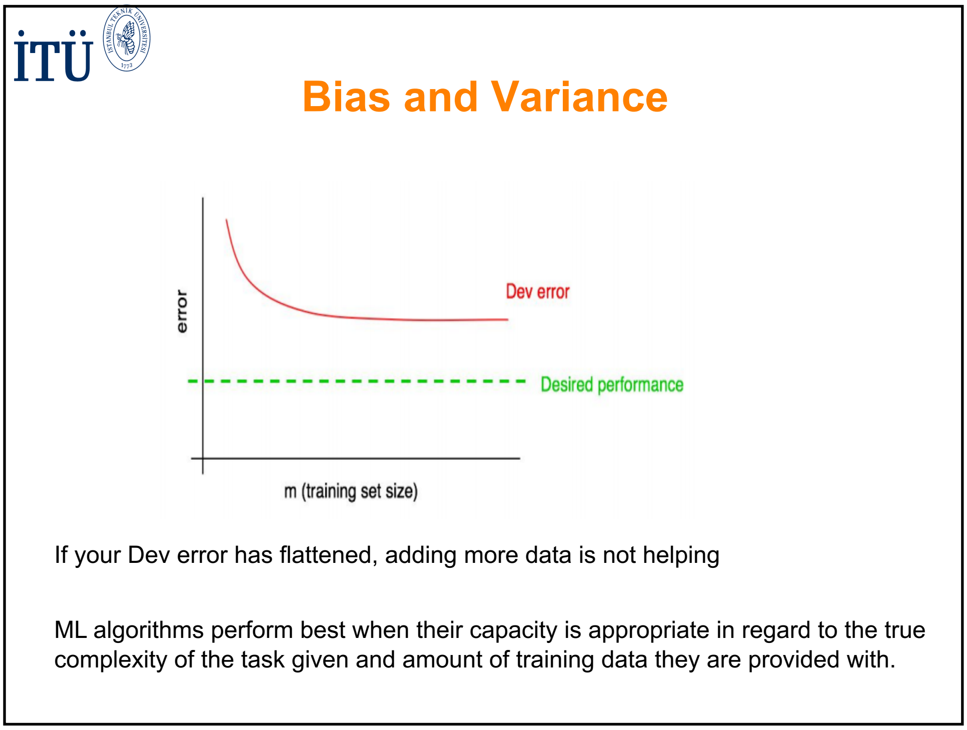
Figures: Andrew Ng, Machine Learning Yearning Book



Bias and Variance

You might have an idea for desired performance: i. human level performance; ii. An intuituion from your earlier development on this dataset

19

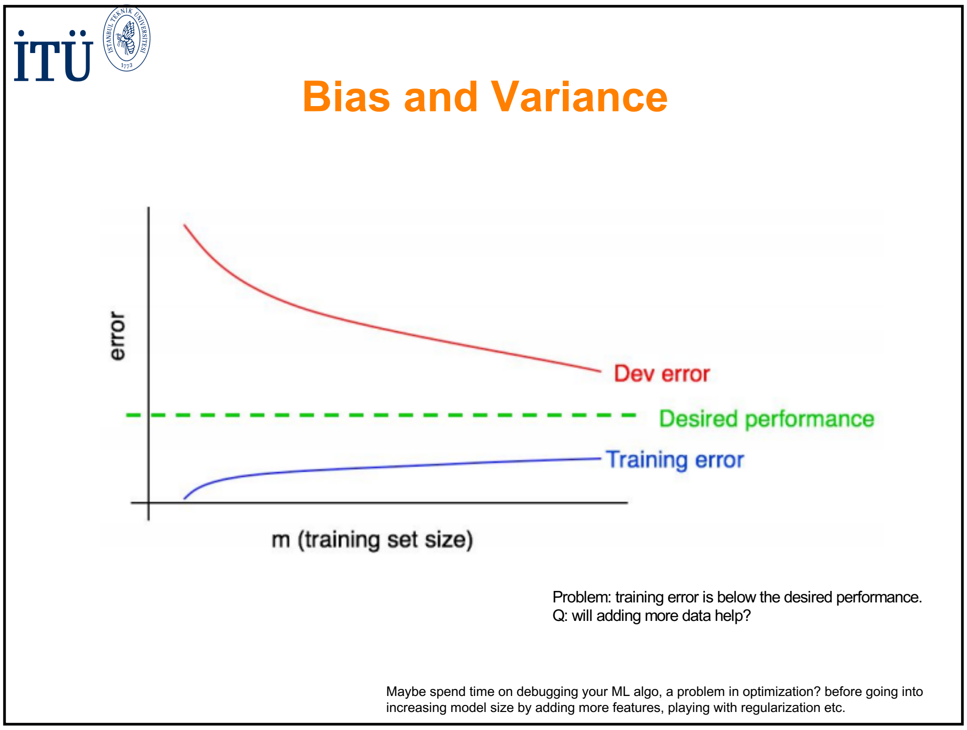


Bias and Variance

If your Dev error has flattened, adding more data is not helping

ML algorithms perform best when their capacity is appropriate in regard to the true complexity of the task given and amount of training data they are provided with.

20

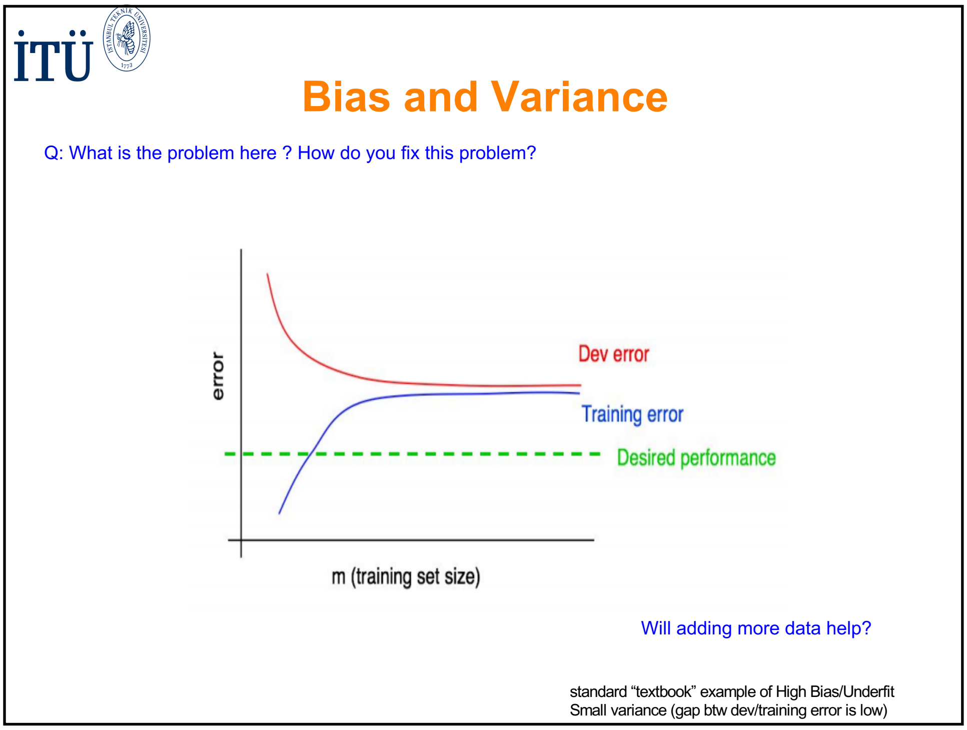


Bias and Variance

Problem: training error is below the desired performance.
Q: will adding more data help?

Maybe spend time on debugging your ML algo, a problem in optimization? before going into increasing model size by adding more features, playing with regularization etc.

21



Bias and Variance

Q: What is the problem here ? How do you fix this problem?

Will adding more data help?

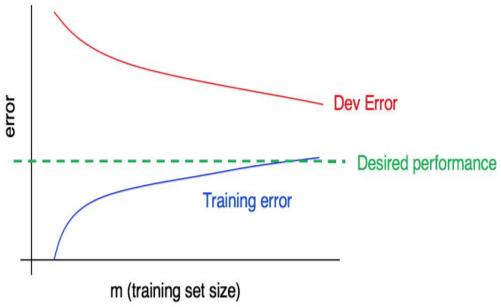
standard “textbook” example of High Bias/Underfit
Small variance (gap btw dev/training error is low)

22

İTÜ

Bias and Variance

Q: What is the problem here ? How do you fix this problem?



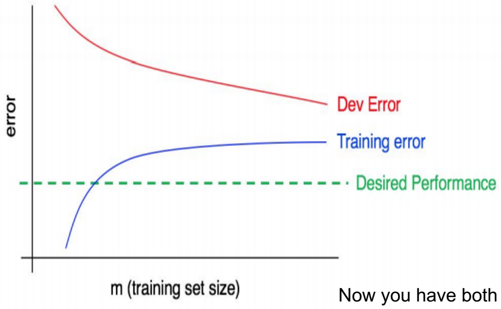
Will adding more data help?

standard “textbook” example of Low Bias (High Variance) /Overfit

23

İTÜ

Bias and Variance

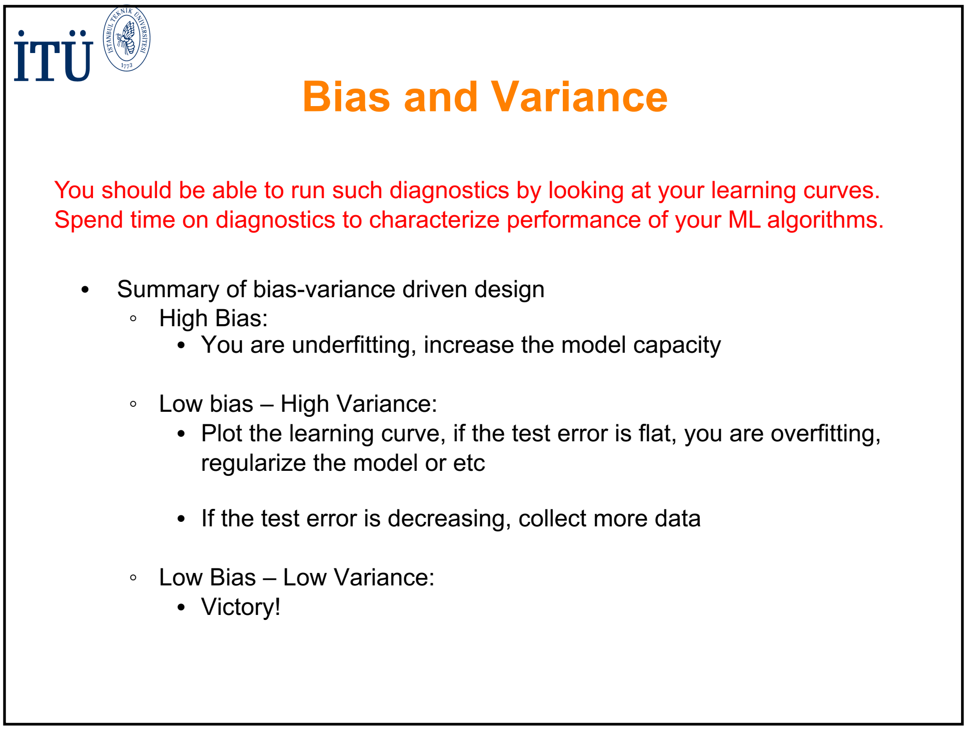


Now you have both large training and dev error.
Both high bias and high variance

Find a way to reduce both bias and variance.

In practice, try to overfit first during training, then start decreasing the capacity of your model while monitoring the dev (val) error

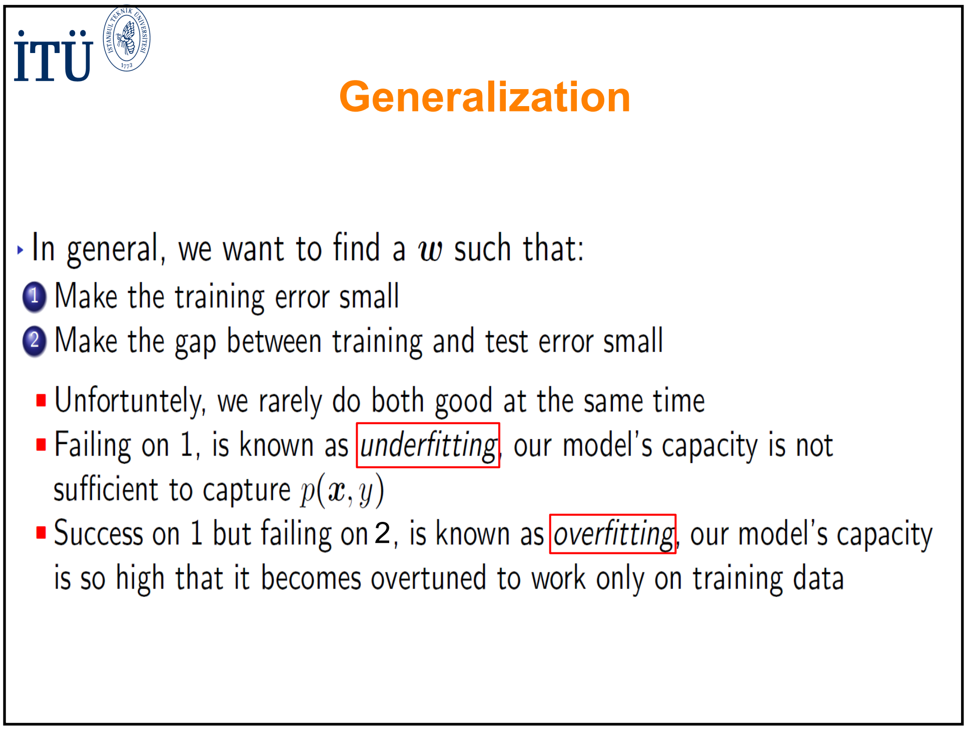
24



Bias and Variance

You should be able to run such diagnostics by looking at your learning curves.
Spend time on diagnostics to characterize performance of your ML algorithms.

- Summary of bias-variance driven design
  - High Bias:
    - You are underfitting, increase the model capacity
  - Low bias – High Variance:
    - Plot the learning curve, if the test error is flat, you are overfitting, regularize the model or etc
    - If the test error is decreasing, collect more data
  - Low Bias – Low Variance:
    - Victory!



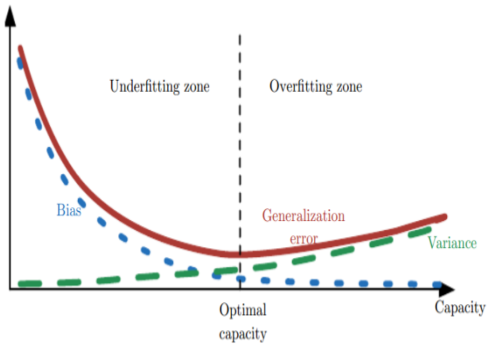
Generalization

- In general, we want to find a $\boldsymbol{w}$ such that:

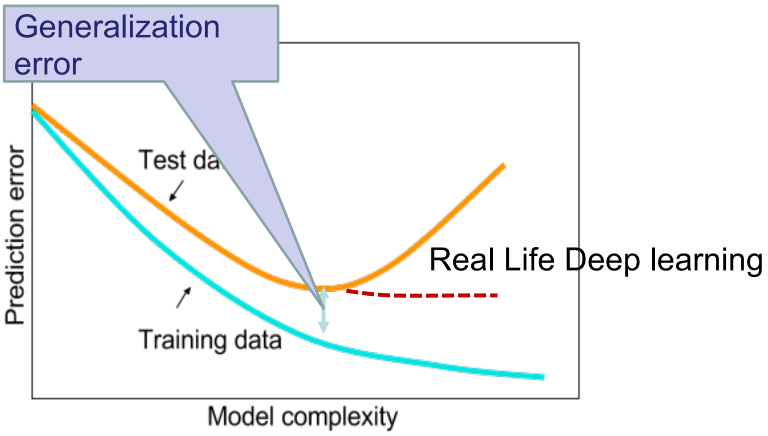
1. Make the training error small
2. Make the gap between training and test error small

- Unfortuntely, we rarely do both good at the same time
- Failing on 1, is known as *underfitting*, our model's capacity is not sufficient to capture $p(\boldsymbol{x}, y)$
- Success on 1 but failing on 2, is known as *overfitting*, our model's capacity is so high that it becomes overtuned to work only on training data

Bias and Variance



textbooks warn us: Large models can “Overfit”



E.g. Classification task on dataset CIFAR10, 50K training samples, train with VGG (20 million parameters), i.e. overparameterize –> works well!!

Longtime belief: SGD + regularization eliminates “excess capacity” of the net

Theoretically understanding deep learning, S. Arora

The ”modern” regime in bias-variance trade-off

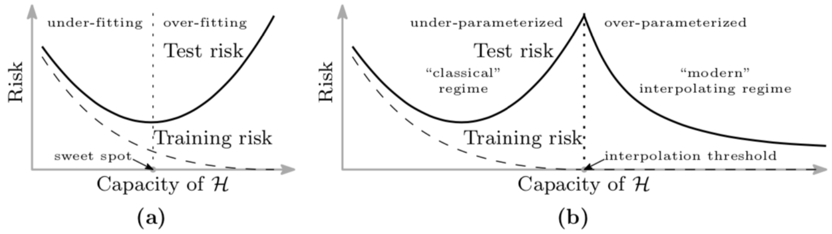
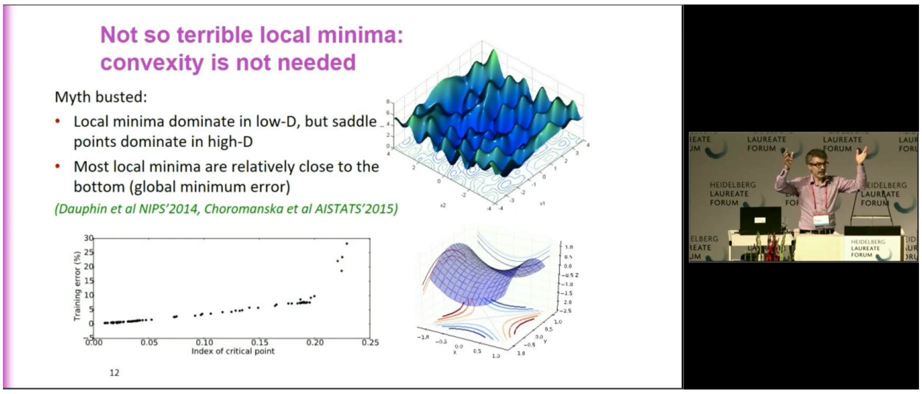


Figure 1: **Curves for training risk (dashed line) and test risk (solid line).** (**a**) The classical *U-shaped risk curve* arising from the bias-variance trade-off. (**b**) The *double descent risk curve*, which incorporates the U-shaped risk curve (i.e., the “classical” regime) together with the observed behavior from using high capacity function classes (i.e., the “modern” interpolating regime), separated by the interpolation threshold. The predictors to the right of the interpolation threshold have zero training risk.

»Reconciling modern machine learning practice and the bias-variance trade-off» , M. Belkin, D. Hsu, S. Ma, S. Mandal. 2019.

29



Not so terrible local minima: convexity is not needed

Myth busted:

- Local minima dominate in low-D, but saddle points dominate in high-D
- Most local minima are relatively close to the bottom (global minimum error)

(Dauphin et al NIPS’2014, Choromanska et al AISTATS’2015)

Bengio Y, talk Heidelberg Laureate Forum

30