Kjersti Engan, professor, leder of BMDLab , Dept. of electrical engineering and computer science

Training a DNN – Hyperparameter tuning and understanding of results

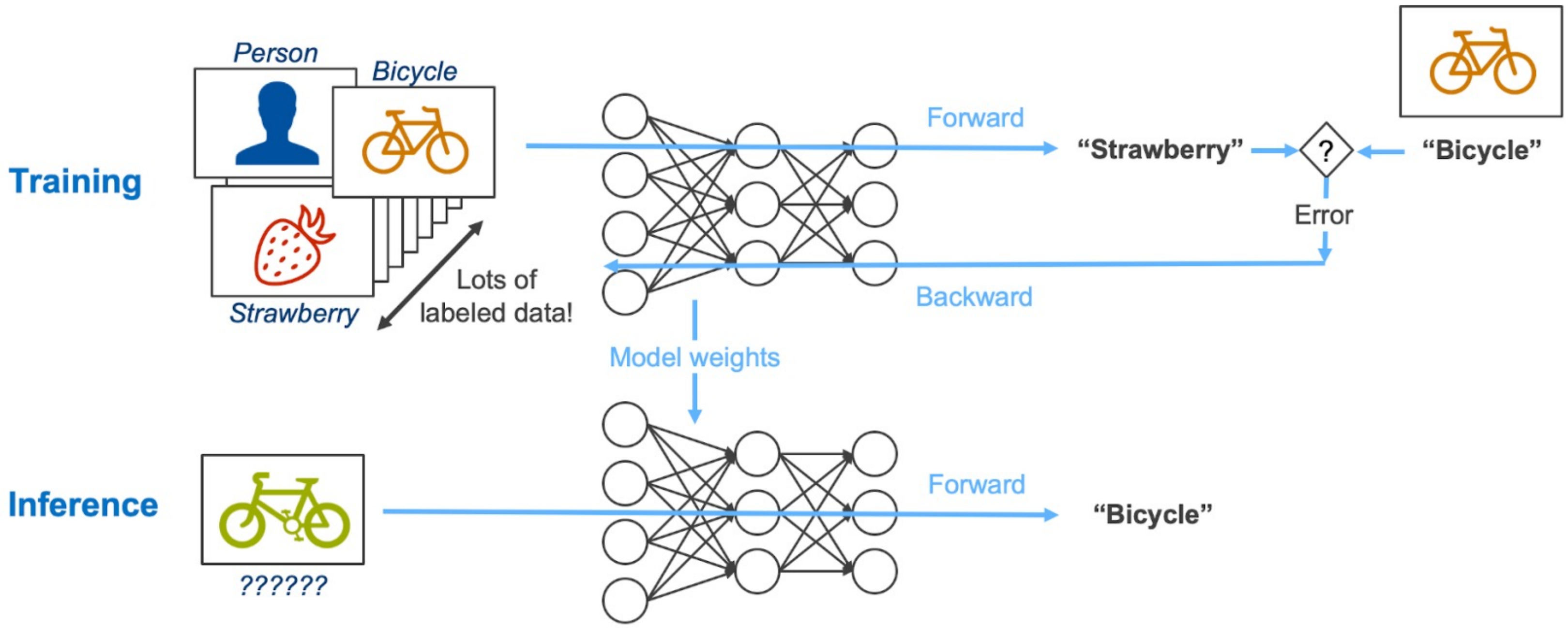
ELE 680 Deep Neural Networks

University of Stavanger

08/07/21

Training a DNN

- Difference between training and inference



https://www.intel.com/content/dam/www/public/us/en/ai/images/training-vs-inference-rwd.png.rendition.intel.web.1648.927.png

Training a DNN - metric

- To train and evaluate we need to choose a (single) evaluation metric

- Optimizing metric : has to be as well as possible on this metric
- Satisficing metrics : has to be better than a threshold on this metric.

- Example
- Max accuracy
- Subject to Running Time < 100 msec

Hyperparameter tuning

- Hyperparameter tuning is choosing a set of optimal hyperparameters for a learning algorithm.
- A hyperparameter is a model argument whose value is set before the learning process begins.
- An improtant key to machine learning algorithms is proper hyperparameter tuning!

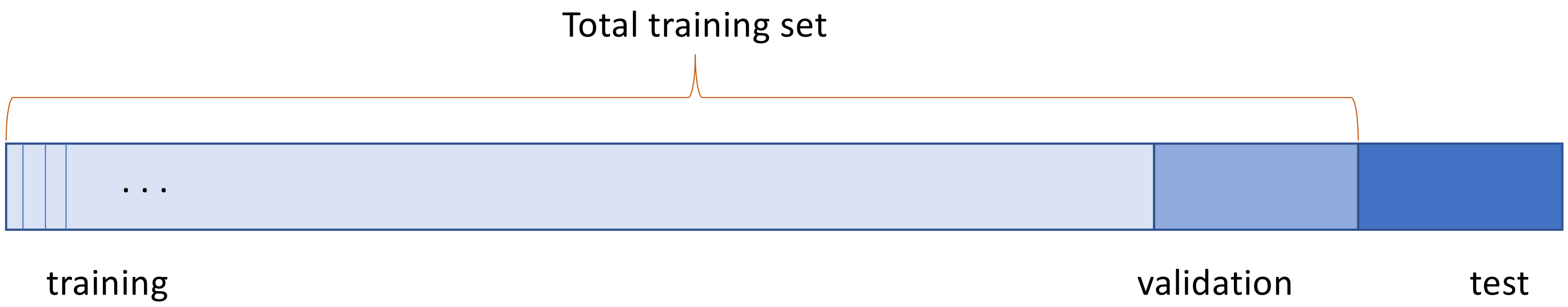
Vinay Setty has already covered parts of this in earlier lecture

Examples of hyperparamters

- K in K-NN (K- nearest neighbours)
- Regularization constant, kernel type, and constants in SVMs
- In Deep Neural Networks
  - Types of layers .. /architecture
  - Number of layers,
  - number of units per layer,
  - Regularization
  - Optimizer
  - Loss function
  - Number of epochs in training
  - etc

Dividing data set

- Typical in NN, your available data needs to be divided into three: training, validation - also called dev (development) set, and test set.



Used to (iteratively) train the weights and paramters of a machine learning algorithm or neural network. Update all the weights and bias terms in a neural network.

To evaluate all different models, througout training and different choices of hyperparamters. Used to select hyperparameters and architectures.

Hold out – to be used «in the end» to show performance of the system/classifier

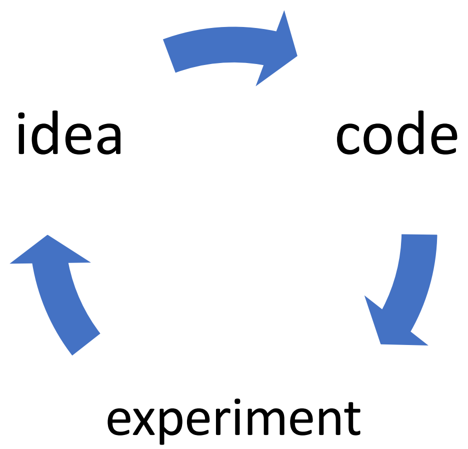
Divide data set

- Some common splits: 80% / 10% / 10%
- Or total training = 80, Test = 20. New split of total training: training 80 , validation 20 (translates into 64 / 16 / 20)
- Really dependent on size of available data sets.

- Set the test set to be big enough to give high confidence in the overall performance of your system.

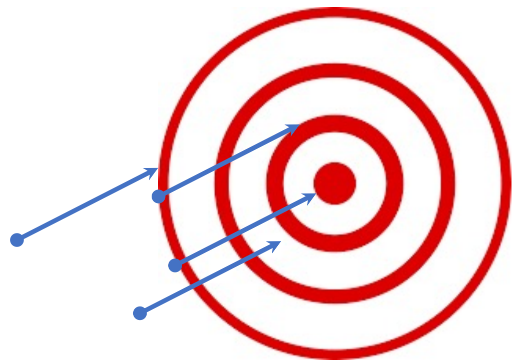
Train/validation/test procedure

1. Randomly initialize each model.
2. Train each model on the training set.
3. Evaluate each trained model's performance on the validation set.
4. Choose the model with the best validation set performance.
5. Evaluate this chosen model on the test set.



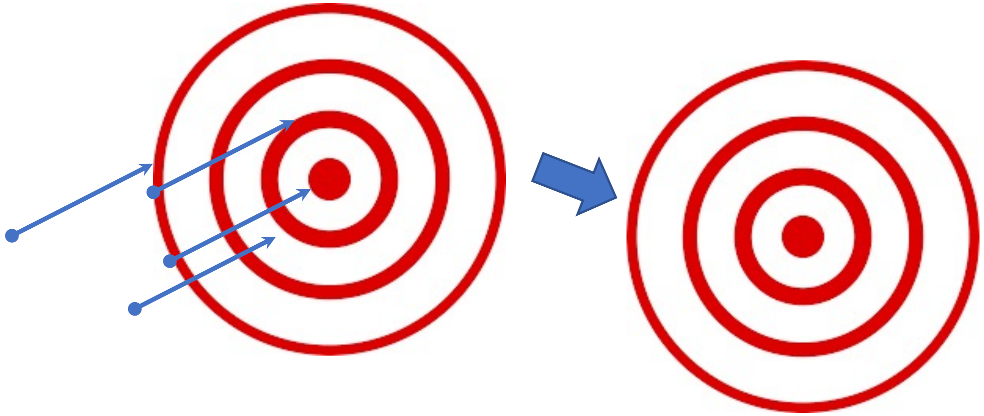
Choice of validation and test set

We optimize based on the validation set. Therefore be sure that the test set and validation set is from similar distribution!



Choice of validation and test set

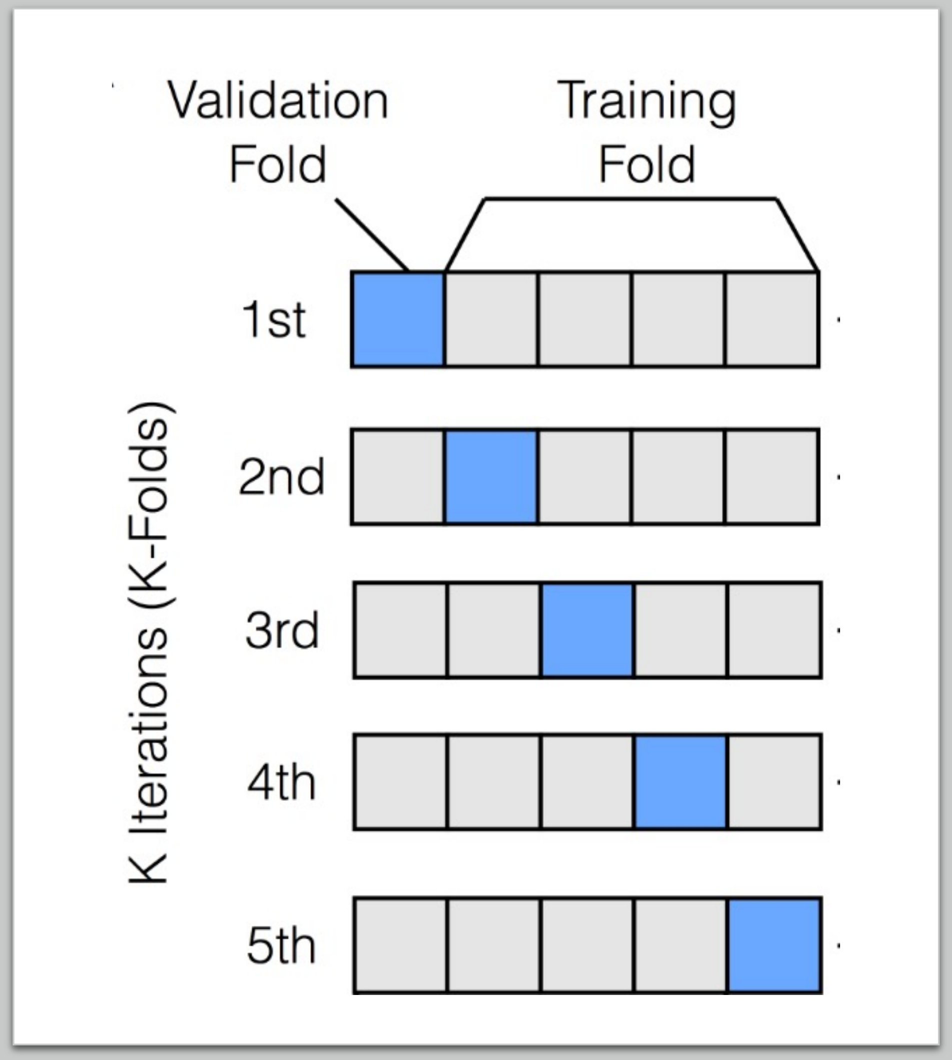
We optimize based on the validation set. Therefore be sure that the test set and validation set is from similar distribution!



- Choose a validation/dev and test set to reflect the data you expect to get in the future and consider important ot do well on

Cross validation

- In classical ML cross-validation is often used – not as much in deep learning because often long training times etc.
- In cross-validation the training+validation set (total training set) is repeatedly split into several training and validation sets.
- The final score is generally the average of all the scores obtained across the k-folds.



https://androidkt.com/pytorch-k-fold-cross-validation-using-dataloader-and-sklearn/

Cross validation

- Cross-validation will provide more stable results, especially in situation with little data, and gives an estimate of model performance. Can be used to decide best model/hyperparamters.
- Chosen model/architecture can be retrained with entire trainingset using best hyperparameters.
- To test the model performance, an additional test data set held out from cross-validation is normally used (sometimes just CV results are reported).

Understanding Results

- **Bayes error rate** is the lowest possible error rate and is analogous to the irreducible error
- **Bayes error** is the very best theoretical mapping function from X to Y, which no machine learning algorithm can ever surpass.
- So why are we often comparing to human level performance?
  - Bayes error rate not known in general
  - Human good at many tasks. If ML method is worse than human you can
    - Get labeled data from humans
    - Get insight from manual error analysis
    - Better analysis of bias / variance

Bias - Variance

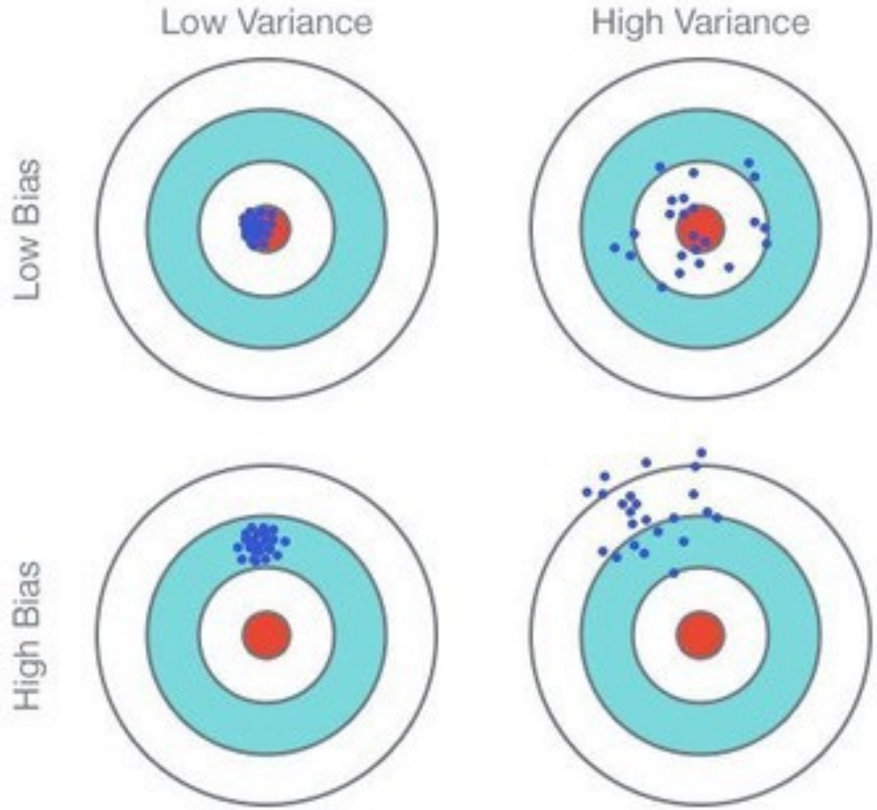


Fig. 1: Graphical Illustration of bias-variance trade-off , Source: Scott Fortmann-Roe., Understanding Bias-Variance Trade-off

Bias - Variance

Bias = 1- Training accuracy.
Variance = Training accuracy – validation accuracy

- Bias describes the model's ability to learn from the training data. A large bias means that the model is having a hard time learning from the training data.
- **Unavoidable bias** is known as the **optimal error rate (Bayes error rate)**.
- **Avoidable bias** is the difference between the optimal error rate and the training error.
- The variance describes how well your model can generalize to data it has not seen yet

By knowing what the bayes or human-level performance is, it is possible to tell when a training set is performing well or not.

Understanding results

Learn WHY your model perform poorly by identifying bias and variance
Learn HOW to improve your model by reducing bias and variance.

Look at learning curves!

Understanding results

bias - high: underfitting (need more data , different/ more complex model/architecture)

variance - high: overfitting (regularisation, L1/L2 reg, dropout, early stopping)

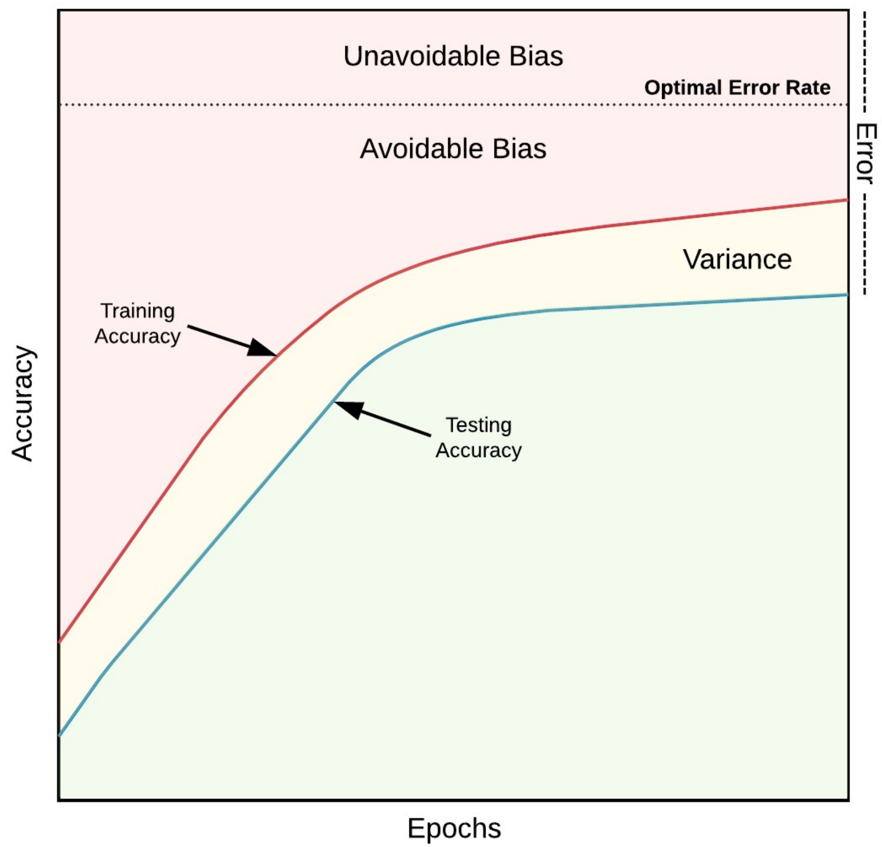
Linear machine learning algorithms often have a high bias but a low variance.

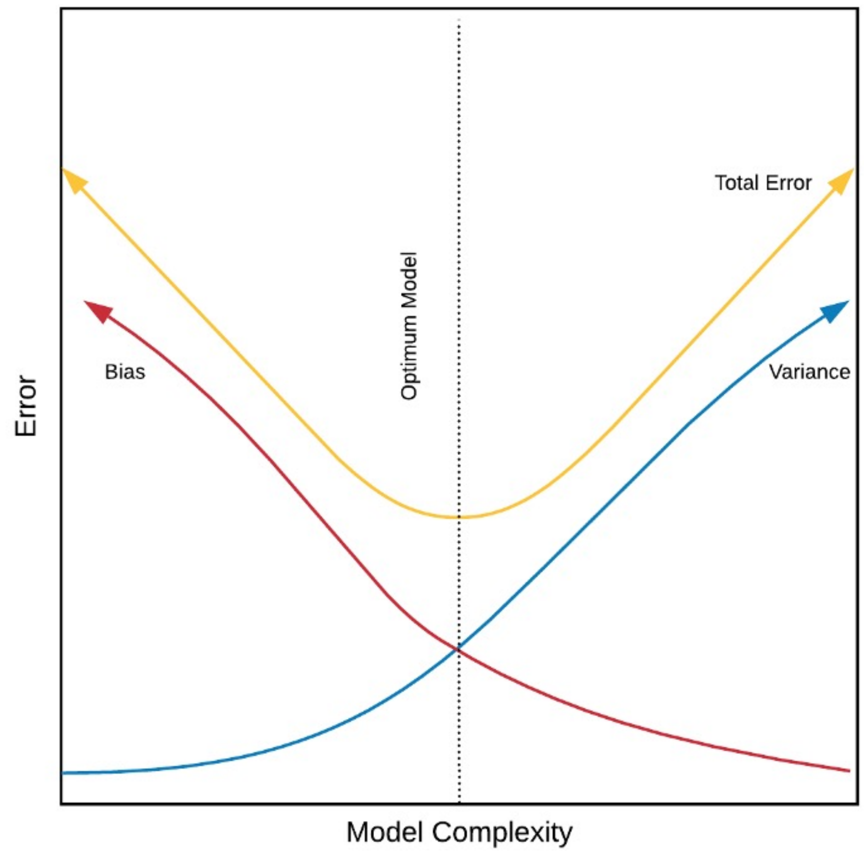
Nonlinear machine learning algorithms often have a low bias but a high variance.

Bias – variance trade-off

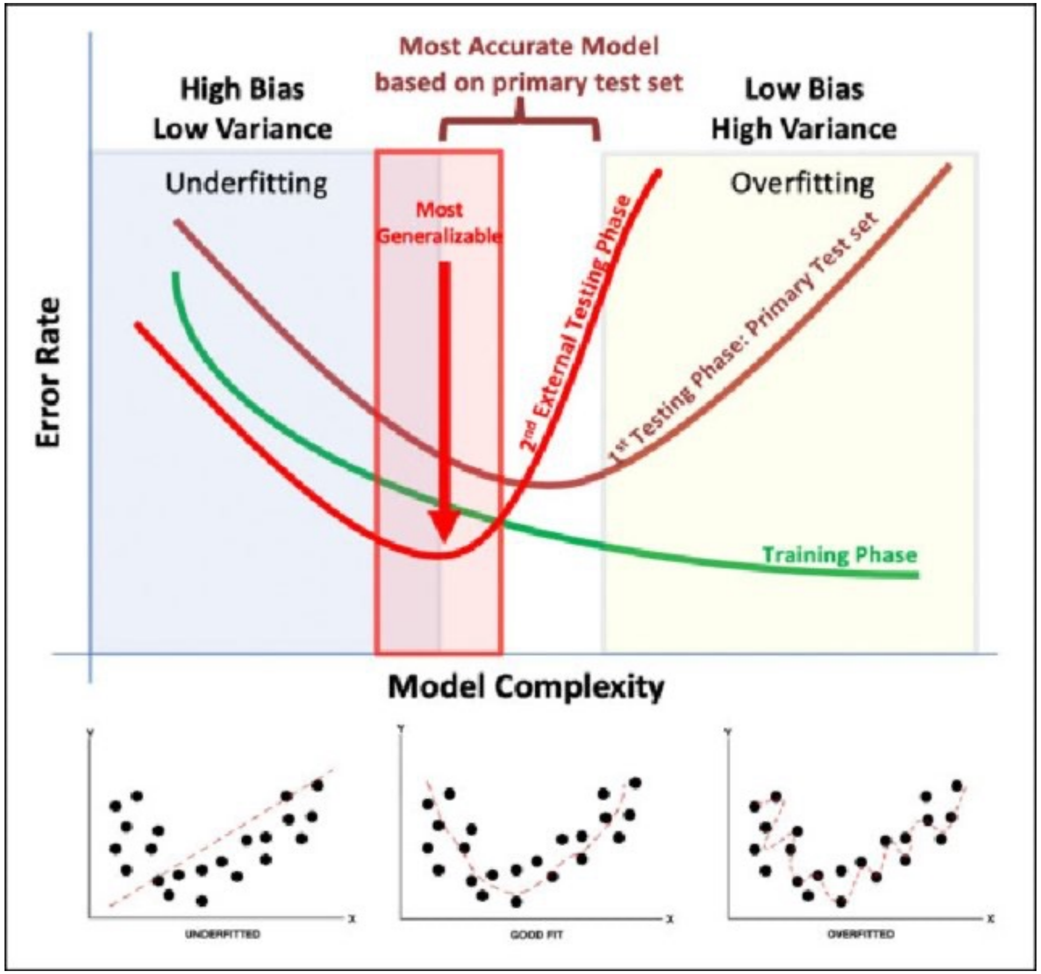
Determined by the complexity of the model and the amount of training data.

The bias – Variance trade-off





https://towardsdatascience.com/two-important-machine-learning-concepts-to-improve-every-model-62fd058916b



Under-fitting: training and validation error are both high (high bias)

Over-fitting: Training error is low (low bias) but validation error is high (high variance)

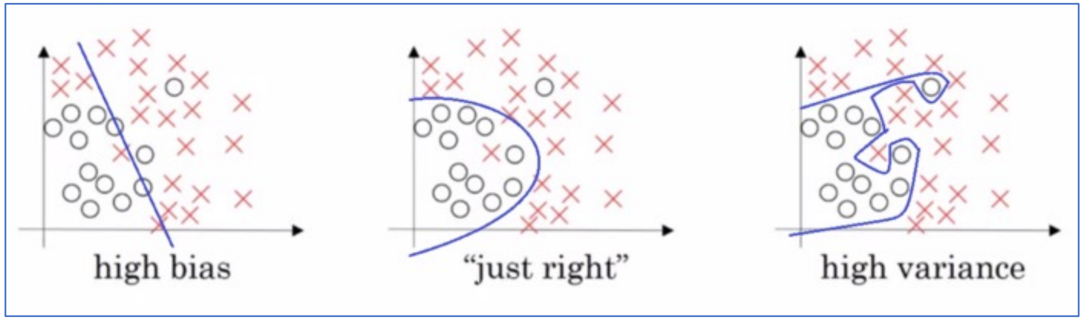
H.Rashidi et.al. «Artificial Intelligence and Machine Learning in Pathology: The Present Landscape of Supervised Methods», DOI: 10.1177/2374289519873088, License CC BY-NC

Reduce avoidable bias

- Increasing the size or complexity of the model
- Reducing regularzation for a model allows the model to fit the training data better. However, less regularization means that your model won’t generalize as well, thus increasing variance -> classic example of Bias vs. Variance tradeoff.

Reduce variance

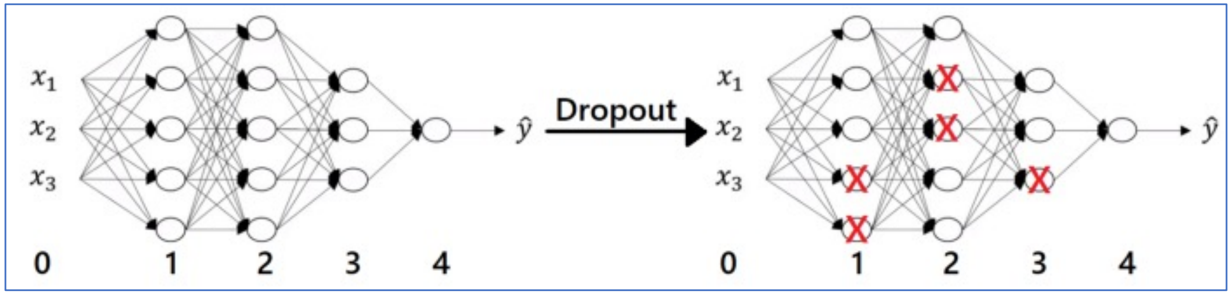
- Adding more data is the simplest way to, *almost* always, increase your model's performance.
- Decreasing the model size will help reduce overfitting on the training data.
- Reducing the dimensionality of your data set by removing features that are not needed is a way to reduce the variance of your model.

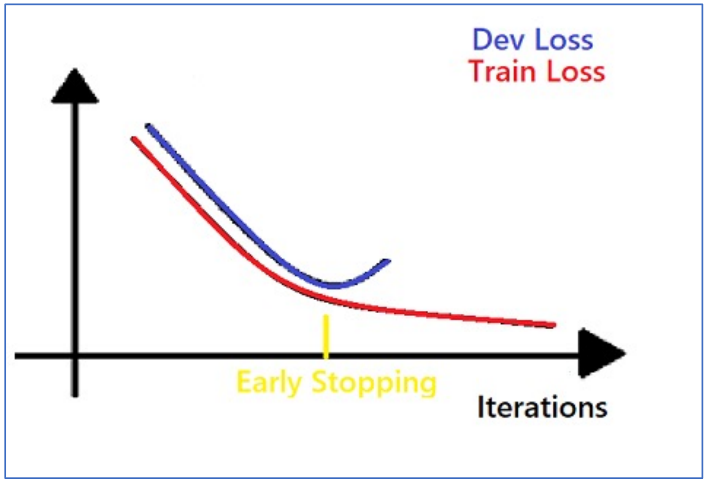


https://towardsdatascience.com/improving-deep-neural-networks-b5984e29e336

Reduce variance

- Adding regularization (L1,L2, dropout) prevents the model from overfitting on the data. Although this reduces variance, it will always increase bias.
- Early stopping





https://towardsdatascience.com/improving-deep-neural-networks-b5984e29e336

- These techniques can change **both** bias and variance.
  1. Layer activation functions (tanh, relu, sigmoid, …)
  2. What the model is learning (ANN, CNN, RNN, KNN, …)
  3. How the model is learning (Adam, SGD, RMSprop, …)
  4. Change other hyperparameters (Learning rate, image size, …)
- Adding new features to the training data can give more information to the model that it can use to learn from.

References and reading

- https://www.intel.com/content/www/us/en/artificial-intelligence/posts/deep-learning-training-and-inference.html
- https://en.wikipedia.org/wiki/Training,_validation,_and_test_sets
- https://towardsdatascience.com/two-important-machine-learning-concepts-to-improve-every-model-62fd058916b
- https://machinelearningmastery.com/gentle-introduction-to-the-bias-variance-trade-off-in-machine-learning/
- https://towardsdatascience.com/improving-deep-neural-networks-b5984e29e336
- https://en.wikipedia.org/wiki/Bias–variance_tradeoff
- https://medium.com/@mp32445/understanding-bias-variance-tradeoff-ca59a22e2a83