Machine Learning Basics Continued

13 January 2026

Alex Lyman

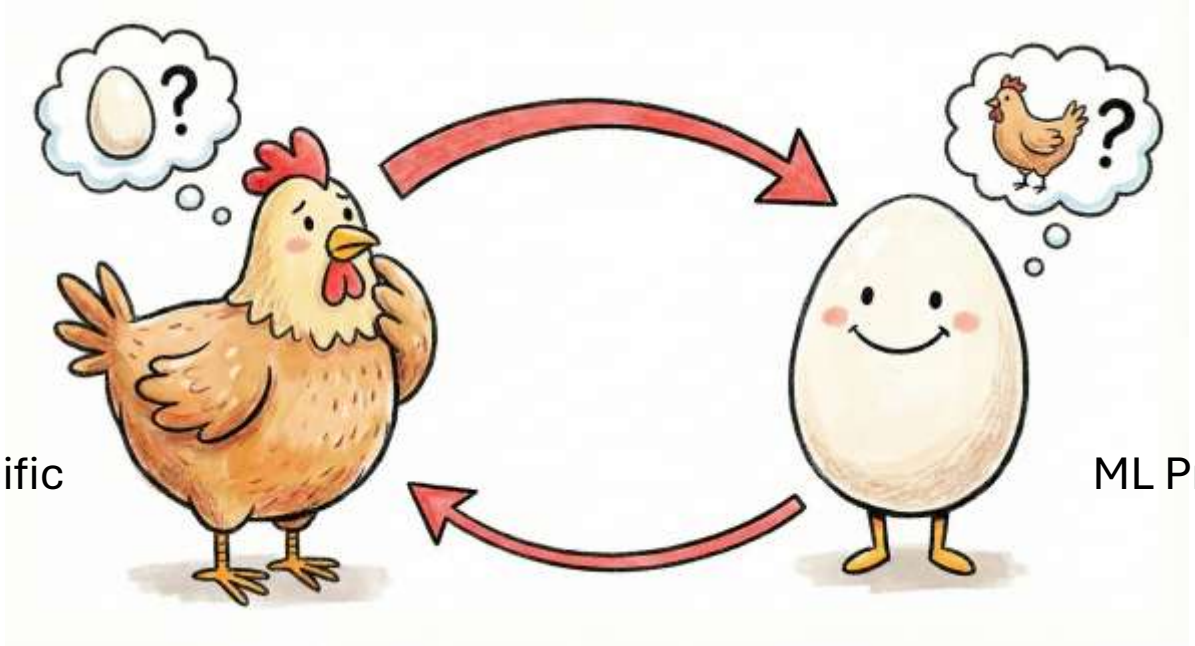
Reminders

First Lecture - Overwhelming

- The first lecture was kind of overwhelming.
- That's *okay*. This is a class full of new material, and you'll probably feel overwhelmed sometimes.
- As an introductory course, you should expect new material almost every class/reading.
- This course covers a lot of stuff, and I don't get as much time as I wish I had per topic.
- Machine Learning is hard to learn. (Applied skill)

Learning Machine Learning

Model-Specific Information



ML Principles

- If you don't have model-specific examples, ML principles are hard to understand in abstract.
- If you don't understand ML principles, it's hard to understand the models.
- That's why we do so much hands-on and the reading is so critical.

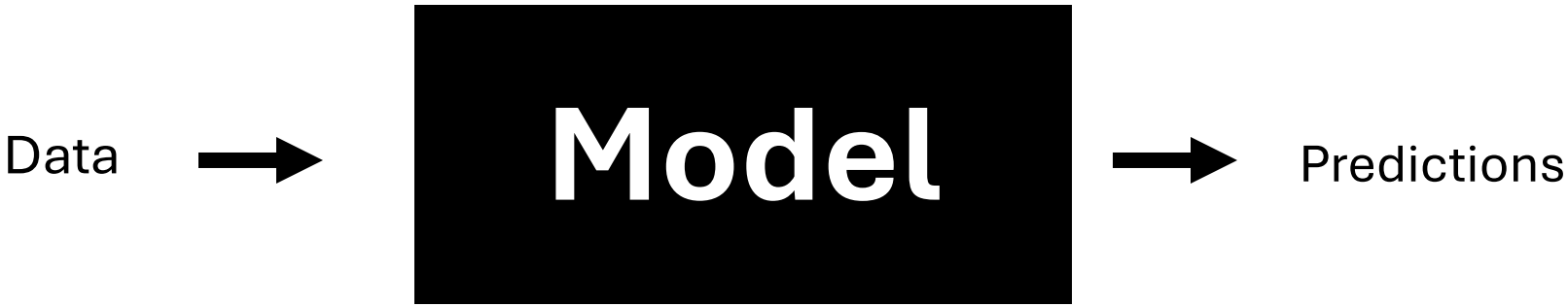
Last Lecture

- Last lecture, we went over a lot of stuff.
- This was me trying to expose you to some of the pillars of machine learning.
- We will go over *most* of that again later in the semester.

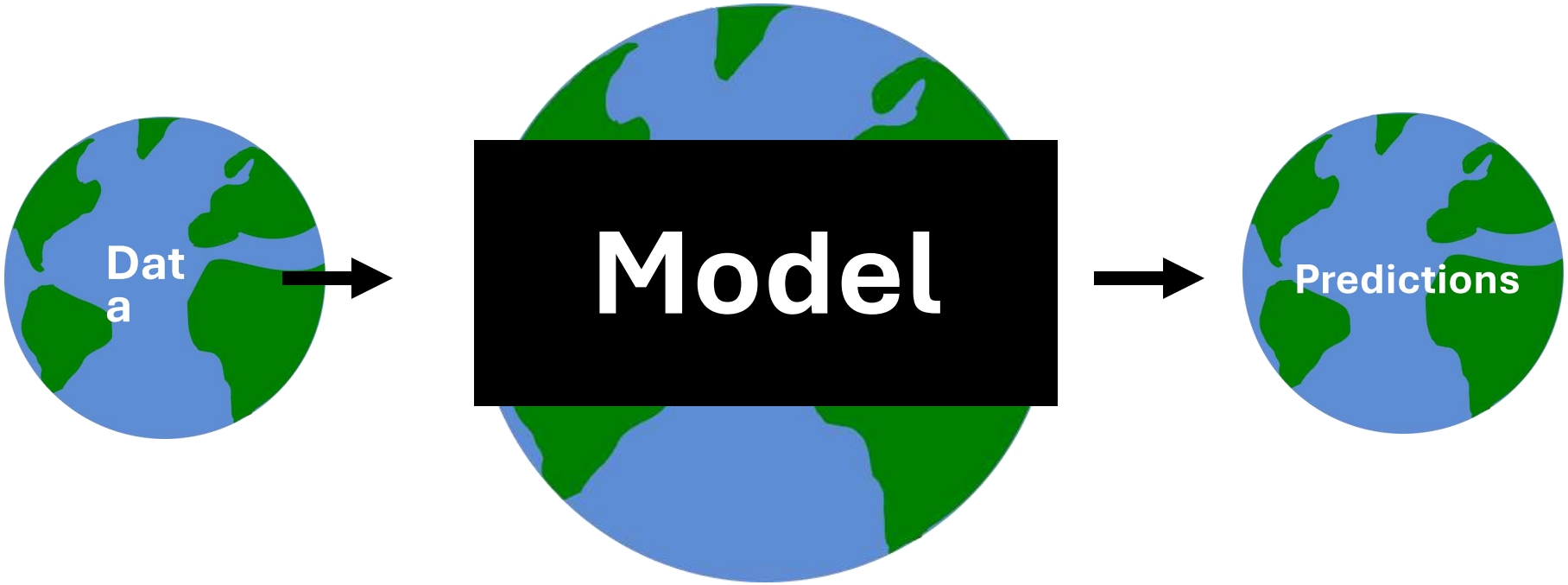
Review of Last Time

- **Machine learning** uses **algorithms** and **models** to make predictions and discover patterns in data.
- A **model** is a mathematical function for describing the relationship between inputs and outputs. We use models to make novel **predictions**.
- It's okay if you can't really imagine what a **model** is right now.

Machine Learning Review



Machine Learning Review



Review of Last Time

- **Models** learn from **data**
- This means that data is *really important*
  - Types of data
  - Importance of visualizing, transforming data
- **Models** make **predictions**
- We **evaluate** our model based on its predictions
  - Accuracy
  - Precision
  - Recall
  - F1
  - (We'll cover these more in depth later)
- Types of Machine Learning
  - Supervised (learn relationship between inputs and outputs)
  - Unsupervised (find structure in data)
  - Reinforcement (agent tries to maximize reward function)

You Won't Understand Everything at Once

Ye cannot bear all things now; nevertheless, be of good cheer, for I will lead you along.
Doctrine & Covenants 78:18

I will give unto the children of men line upon line, precept upon precept, here a little and there a little.
2 Nephi 28:30

And see that all these things are done in wisdom and order; for it is not requisite that [you] *should run faster than* [you have] *strength. And again, it is expedient that* [you] *should be diligent, that thereby* [you] *might win the prize; therefore, all things must be done in order.*[8]
Mosiah 4:27

It has been my experience that the Lord gives us just as much as we have the strength to handle, plus a little extra, so that we can increase our faith and strength. Just as with all of Heavenly Father's creations, you were designed to grow and progress. You were not meant to stay the way you are. Change and improvement are built into your eternal DNA.

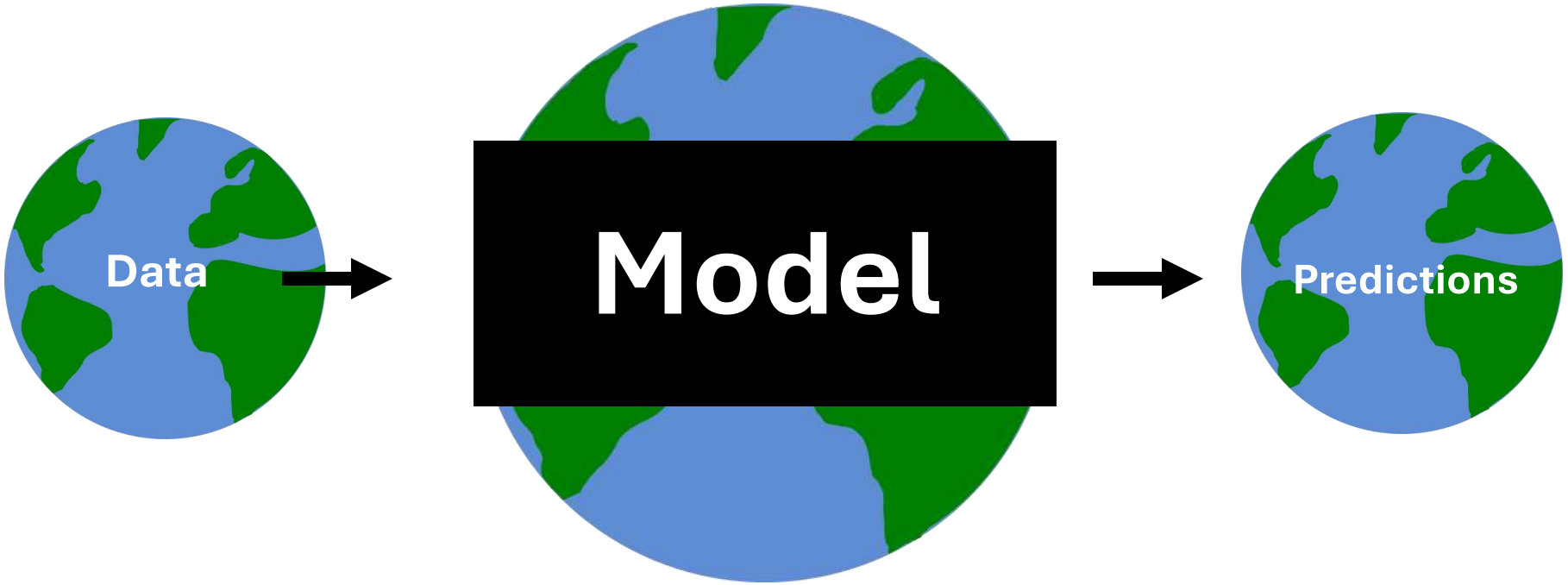
Jan E. Newman – 2nd Counselor, Sunday School General Presidency
BYU Devotional – Feb 6, 2024

How do we feel now?

Couple More Basics

Generalization and Overfitting

Generalization



Want a model to "generalize" well to _____ data?

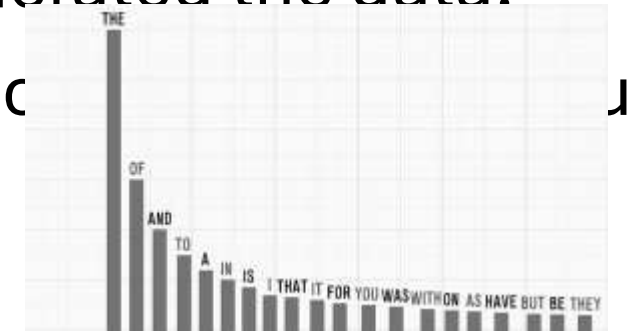
unseen

(Want "high generalization accuracy" or "low generalization error")

Generalization

- i.i.d. assumption: training and test examples are independent and identically distributed (drawn from the same joint probability distribution, P(X, y))

- We want to fit a model to our training set, that captures/reflects the distribution that generated the data.
  - Example: LLMs c ution of natural language



Overfitting

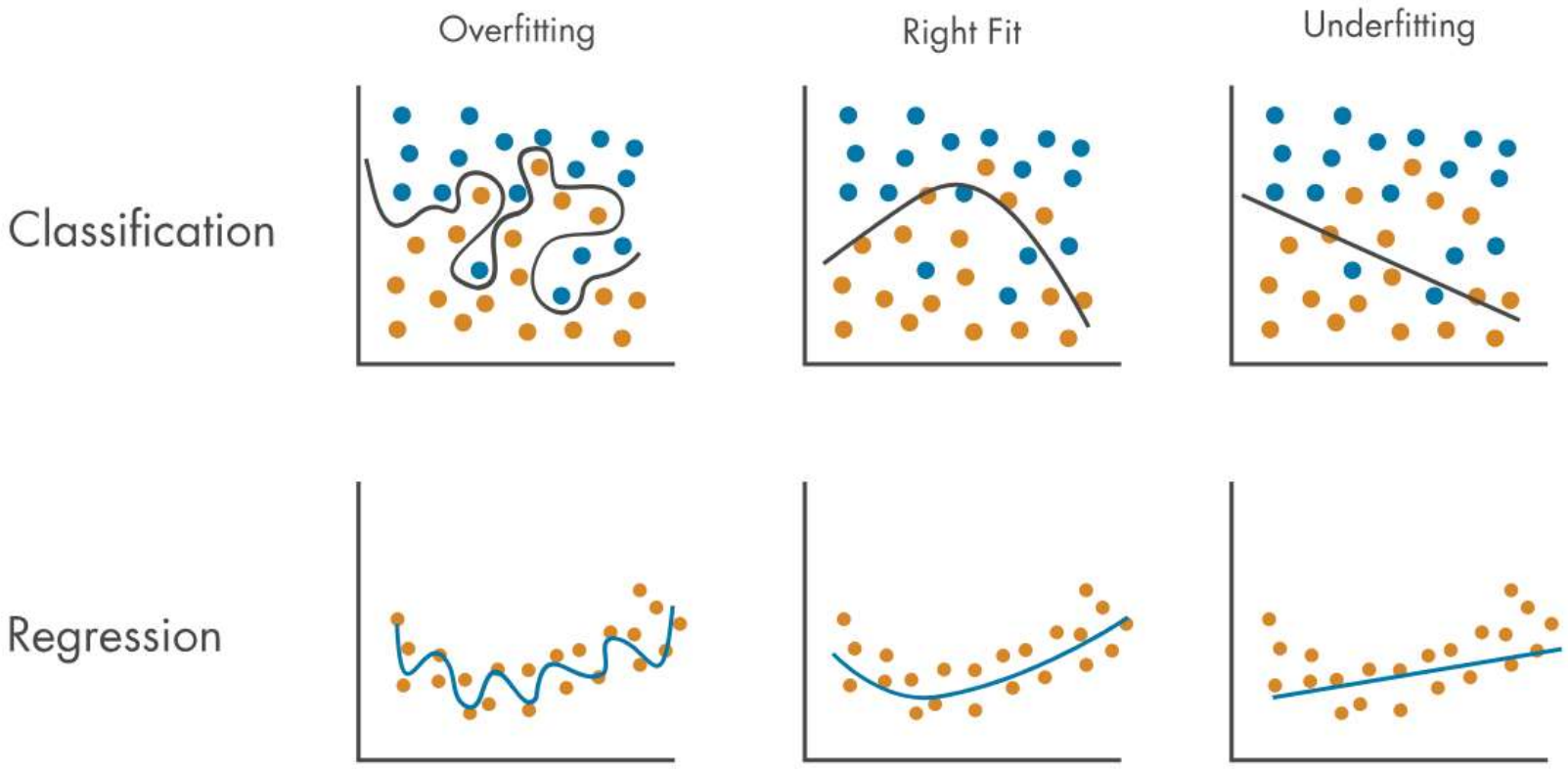


Image Credit: Mathworks

Typically try to learn a model just complex enough to do well and no more complex than that

Avoiding Overfit

Regularization: *any modification we make to learning algorithm that is intended to reduce its generalization error but not its training error*

- Occam's Razor – William of Ockham (c. 1287-1347)
  - Favor simplest explanation which fits the data
- General Key: Focus on patterns/rules that really matter and ignore others
- More Training Data (vs. overtraining on same data)
  - Data set augmentation – Fake data, Can be very effective, Jitter, but take care… (discuss on next slide)
  - Denoising – add random noise to inputs during training – can act as a regularizer
  - Adding noise to models. e.g. (Random Forests, Dropout , discuss with ensembles)

Data Augmentation Example

- Challenge: Train a tiny language model on 10,000 words (the number of words a baby hears by the time it learns to talk.
- Modern LLMs are trained on trillions of words (1,000,000,000,000)
- How do you augment natural language data?

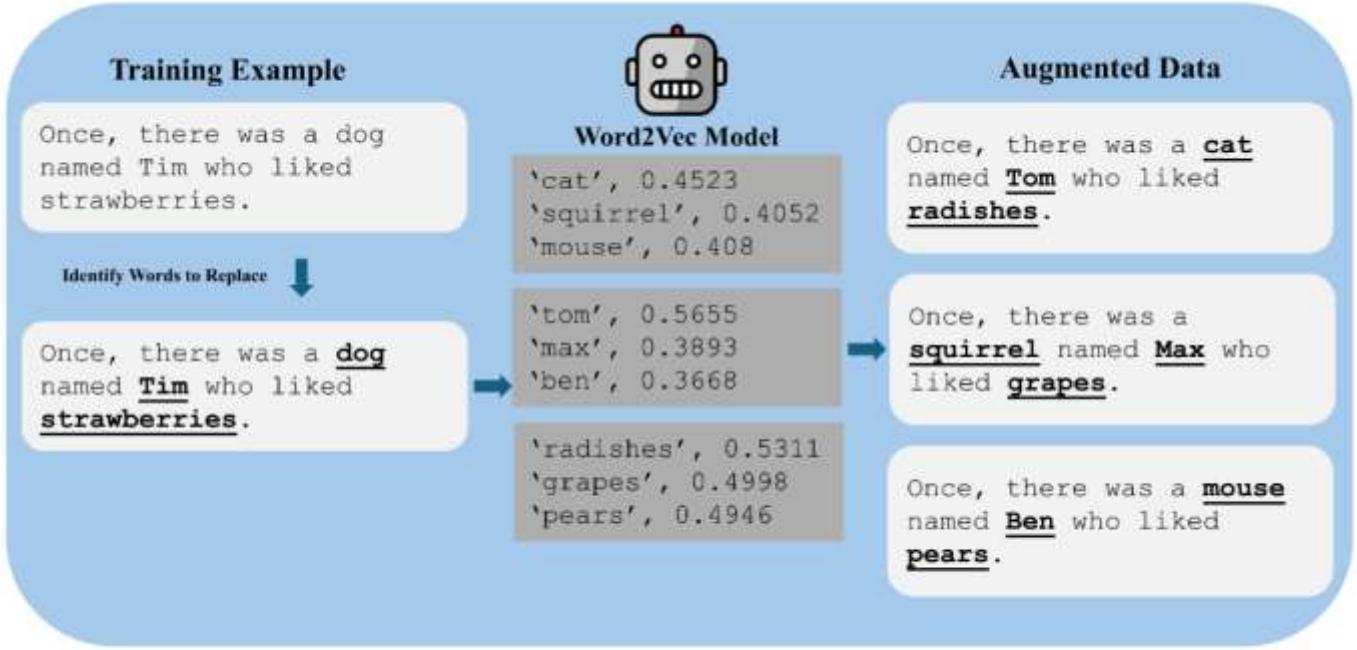


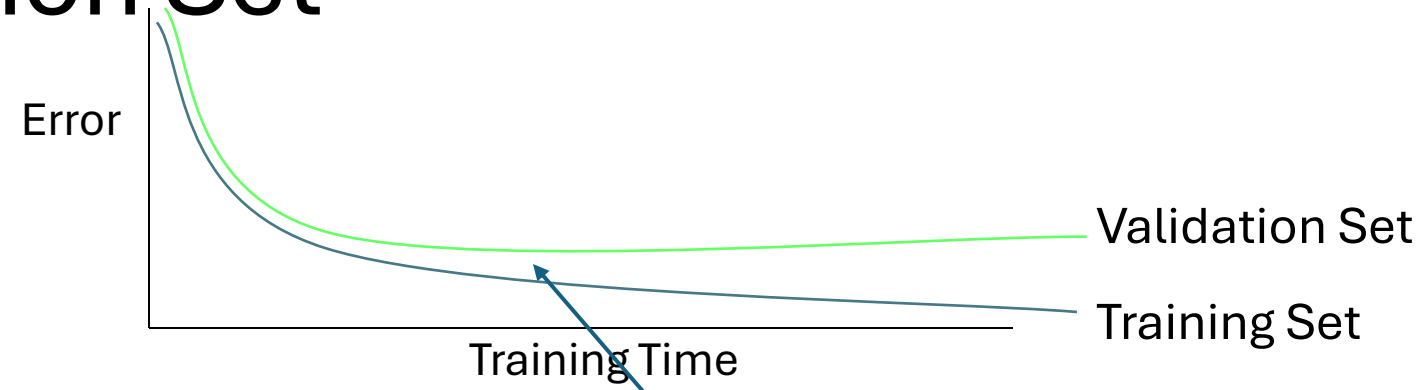
Figure 1: Illustration of the data augmentation technique.

Avoiding Overfit

Regularization: *any modification we make to learning algorithm that is intended to reduce its generalization error but not its training error*

- Occam's Razor – William of Ockham (c. 1287-1347)
  - Favor simplest explanation which fits the data
- General Key: Focus on patterns/rules that really matter and ignore others
- More Training Data (vs. overtraining on same data)
  - Data set augmentation – Fake data, Can be very effective, Jitter, but take care… (discuss on next slide)
  - Denoising – add random noise to inputs during training – can act as a regularizer
  - Adding noise to models. e.g. (Random Forests, Dropout , discuss with ensembles)
- *Early Stopping* – Very common regularization approach: Start with simple model (small parameters/weights) and stop training as soon as we attain good generalization accuracy
  - Common early stopping approach is to use a validation set

Early Stopping/Model Selection with a Validation Set



- Why might this happen?
- There is a different model *h* after each epoch
- Select a model in the area where the validation set accuracy flattens
- Keep *bssf* (Best Solution So Far). Once you go *w* epochs with no improvement stop and use the parameters at the *bssf w* epochs ago.
- The validation set comes out of training set data
- Still need a separate test set to use after selecting model *h* to predict future accuracy

Bias & Variance

One Definition for Inductive Bias

- Inductive Bias: The set of assumptions or preferences an algorithm uses to generalize beyond its training data.

  This guides the model to prefer certain solutions over others when multiple possibilities fit the observed data. This enables learning by restricting the vast search space of potential functions.

Sometimes just called the *Bias* of the algorithm (don't confuse with the bias weight of a neural network).

Bias & Variance

- Learning involves the ability to generalize from past experience to deal with new situations
- If we are only consistent with what we have seen, we can't generalize beyond our experience
  - All the cats I have seen are yellow.
  - What happens when I see a black cat?
- Bias is choosing one generalization over another
  - (classify animals based on shape, not color)
- Variance is how varied my choices are
  - (How lose is my definition of cat?)

Intuition

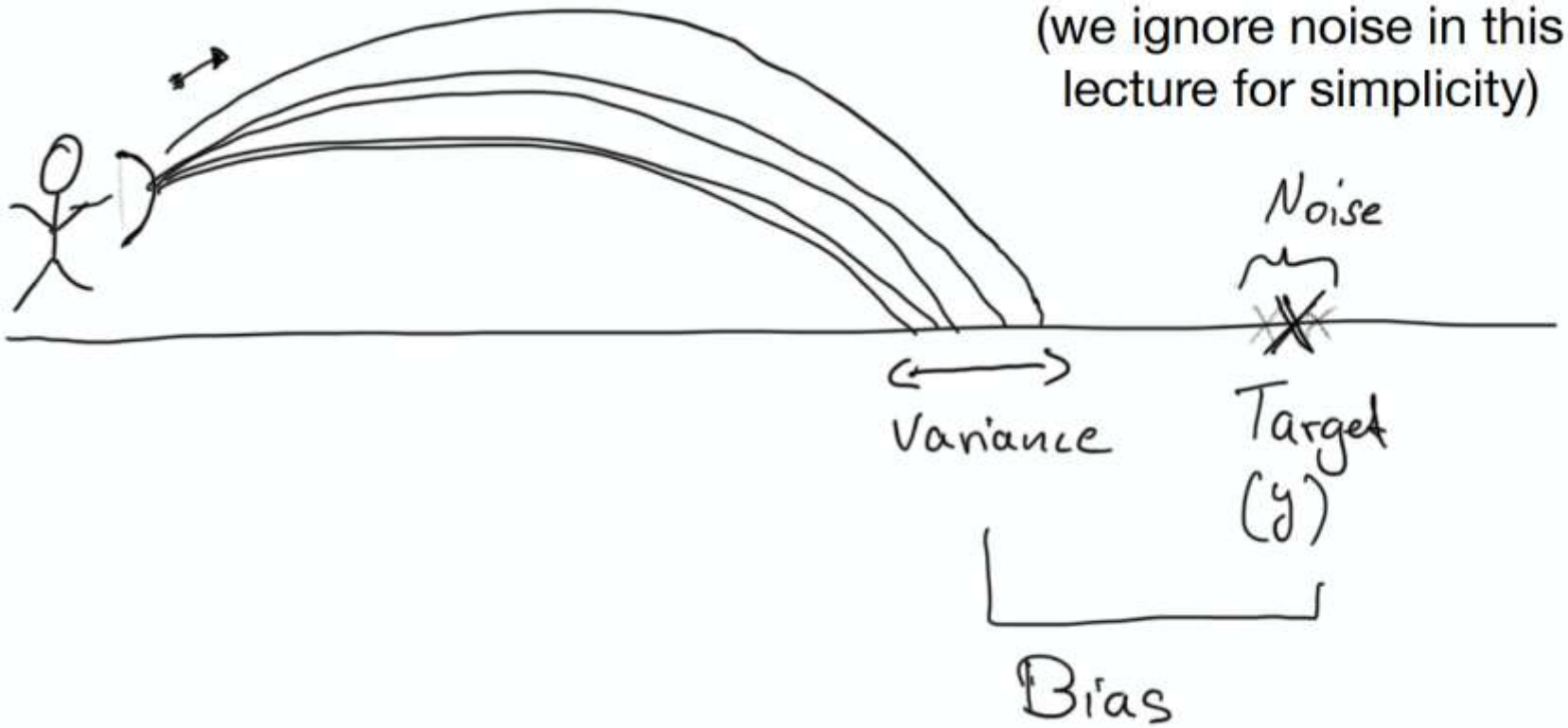
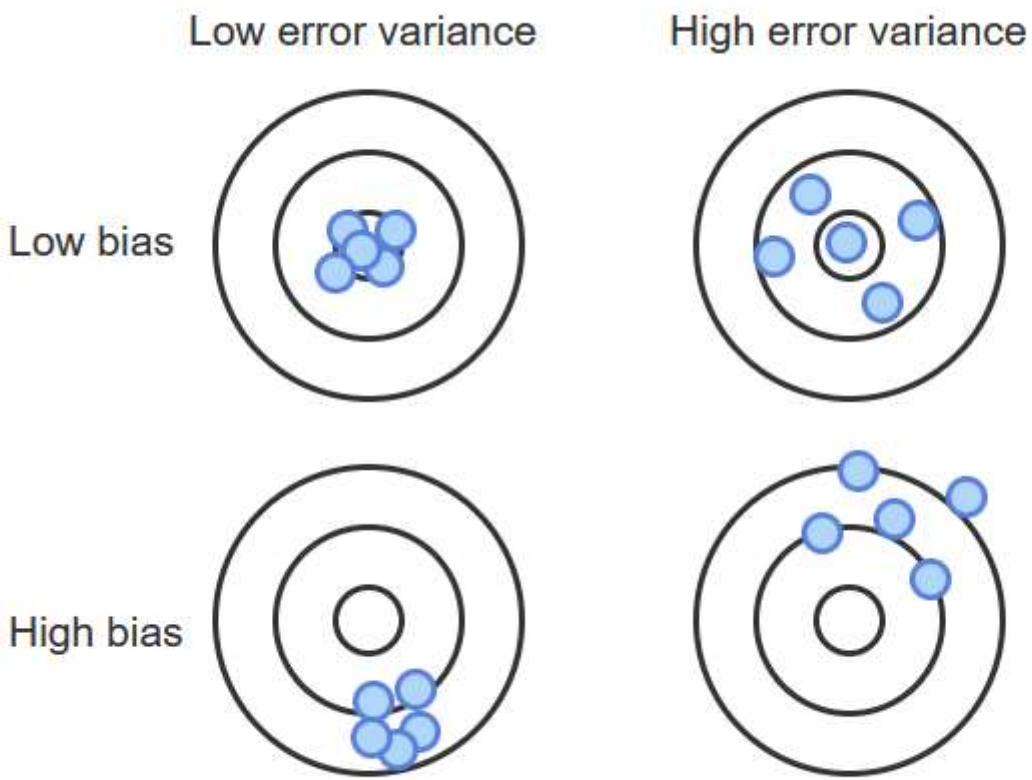


Image Credit: Sebastian Raschka

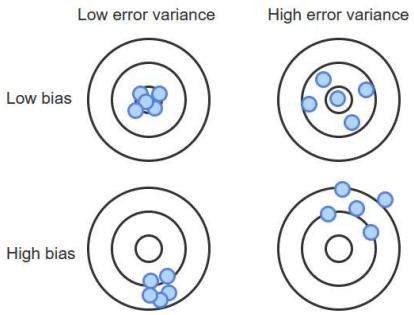
Bias & Variance



Bias vs. Variance

- If there is **no bias**, the outcome of the learner is highly dependent on the training data, and thus there is **much variance** among the models induced from different sets of observations
  - Each learner memorizes (overfits)
- If there is a **strong bias**, the outcome of the learner is much less dependent on the training data, and thus there is **little variance** among induced models
  - Learner ignores observations
- Formalized as:
  - **Bias-variance trade-off**

Bias-Variance Trade-off



- Weak/no bias
  - Observed instances are memorized
  - Learner **overfits**
- Strong/extreme bias
  - Observed instances are mostly ignored
  - Learner **underfits**
- Example: fitting arbitrary polynomial vs straight line to sine wave-like observations
- Bias-Variance Error Decomposition
  - All learning algorithms have a bias
    - Decision trees (simplicity), k-NN (similarity), etc.
  - All learning algorithms may be subject to variance in the data
  - Two sources of error
- How can we estimate the bias and variance of an algorithm?
  - Run the algorithm on different random variations of several datasets
  - Examine the errors made for each variation
    - If the algorithm tends to make the **same errors**, then it must have a **strong(er) bias** [one may need a more flexible algorithm]
    - If the algorithm tends to make **random errors**, then it must have a **strong(er) variance** [one may need a less flexible algorithm or more data]