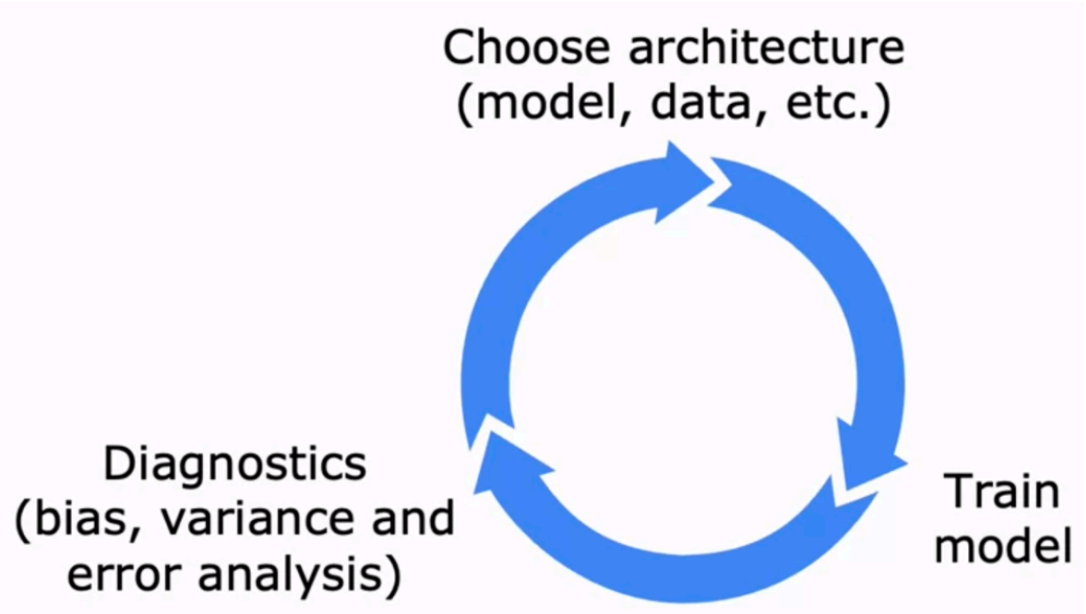
ML development process

Iterative loop of ML development

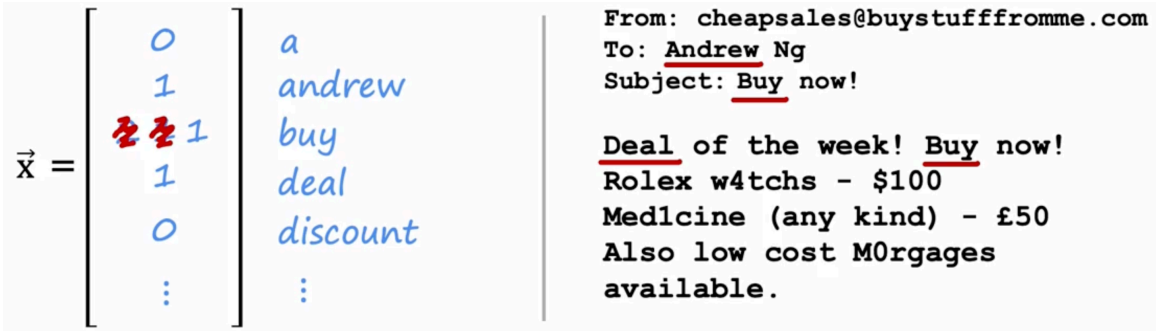
- Begin by deciding the overall architecture, including model selection and data choice.
- Implement and train the model, followed by diagnostics to assess bias and variance.



Example: Building an Email Spam Classifier

Supervised learning: $\begin{cases} x = \text{features of email} \\ y = \text{spam } (1) \text{or not spam } (0) \end{cases}$

Features: list the top $10,000$ words to compute $x_1, x_2, x_3, \ldots, x_{10,000}$



How to try to reduce your spam classifier's error?

- **Collect more** data. E.g., "Honeypot" project.
- **Develop sophisticated features** based on email routing (from email header).
- **Define sophisticated features** from email body.
  E.g., should "discounting" and "discount" be treated as the same word.
- **Design** algorithms to detect **misspellings**.
  E.g., w4tches, medicine, mOrtgage.

Error analysis

$m_{cv} = 500$ examples in cross validation set.

Algorithm misclassifies 100 of them.

Manually examine 100 examples and categorize them based on common traits.

- Pharma: 21
- Deliberate misspellings (W4tches, medicine): 3
- Unusual email routing: 7
- Steal passwords (phishing): 18
- Spam message in embedded image: 5

⇒ **Pharma** and **Steal passwords** are traits that been seen a lot in the misclassified examples.

- Involves manually reviewing misclassified examples to identify common traits or patterns.
- Helps prioritize which issues to address based on the frequency and impact of misclassifications.

Focus Your Efforts:

If you spend a lot of time fixing rare error types (like deliberate misspellings, only 3/100), the overall impact on your model's performance will be small.

Instead, prioritize fixing the most common error types (like pharma spam or phishing).

Inspire Targeted Improvements:

- For pharma spam: Collect more examples of pharma spam or design features that specifically detect pharma-related content.
- For phishing: Add features that analyze suspicious URLs or collect more phishing examples.
- For rare errors: Unless you have extra time, you might not need to focus on these right away.

KEY:

- **Error analysis** helps you decide what to fix next by showing you which mistakes are most common.
- **Prioritize** improvements that will fix the largest number of errors.
- **Don't waste time** on rare errors unless you have already fixed the big ones.
- This process can save you weeks or months of work by focusing your efforts where they matter most.

Adding data

1. **Add More Data of Everything**

- Collect more examples from all categories, regardless of the type. **Example:** The “Honeypot” project.
- **When to use:** When you don’t know which types of errors are most common, or when it’s easy and cheap to get more data.

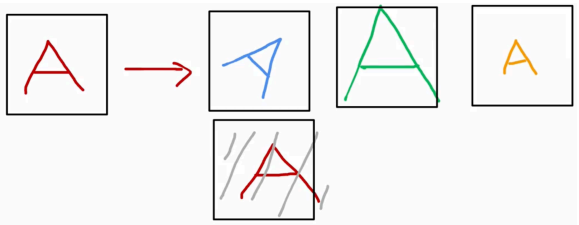
2. **Add More Data of Specific Types (Targeted Data Collection)**

- Focus on collecting more data for the specific types of errors your model struggles with, as identified by error analysis. **Example:** If error analysis shows that “pharma spam” is a big problem, collect more pharma-related spam emails.
    - **How: Eg.,** Go through unlabeled data and find more examples of the problematic type (like pharma spam).
- **Why:** This is often much more efficient and impactful than just collecting more random data.

Beyond Collecting New Data: Data Augmentation

Data Augmentation

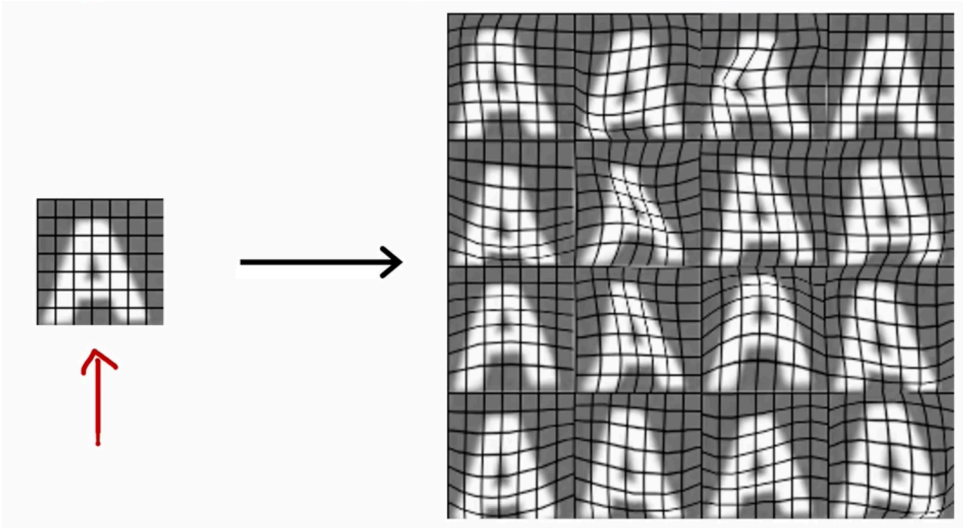
- Create new training examples (x, y) by modifying existing ones. This helps your model learn to handle variations it might see in real-world data, making it more robust.
- **Examples:**
    - For images: Rotate, crop, or change the brightness of a picture.



    - For audio: Add background noise or distort the sound.

Data Augmentation by introducing distortions

- Distortion introduced should be representation of the type of noise/distortions in the test set.
- Usually does not help to add purely random/meaningless noise to your data.



$$x_i = \text{ intensity (brightness) of pixel } i$$

Data Augmentation for speech

Speech recognition example:

Original audio (voice search: "What is today's weather?")

- Noisy background: Crowd
- Noisy background: Car
- Audio on bad cellphone connection

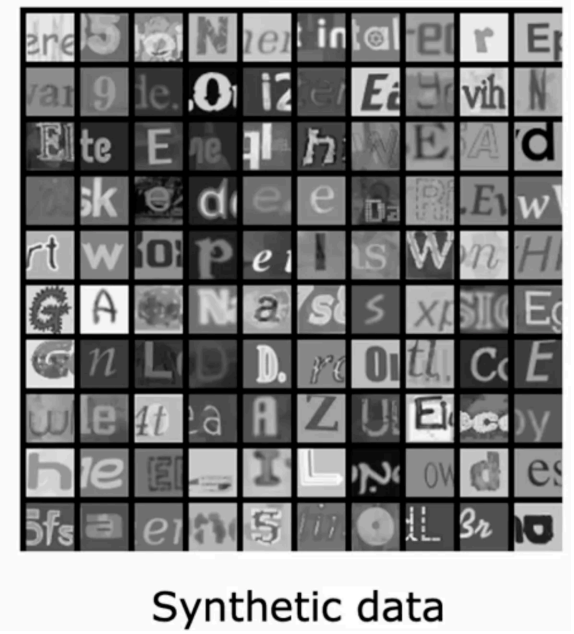
$$x_i \leftarrow x_i + \text{ random noise}$$

Data synthesis

Synthesis: using artificial data inputs to create a new training example.

Artificial data synthesis for photo OCR

Real data



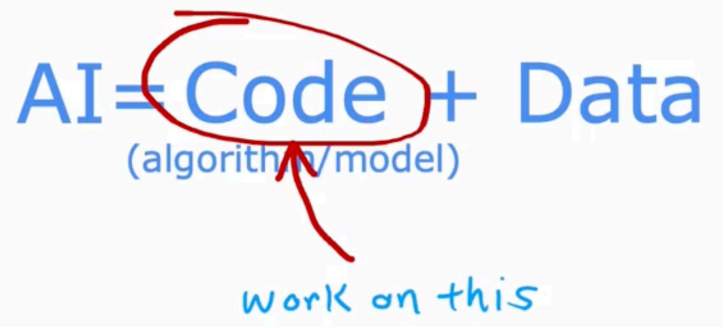
Synthetic data

Engineering the data used by your system

All the techniques seen related to finding ways to engineer the data used by your system.

Conventional model-centric approach:

Using algorithm to improve Code.



Data-centric approach:

Using algorithm to improve Data.

Transfer Learning: using data from different task

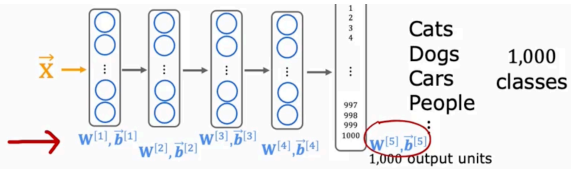
Transfer Learning

Transfer learning is a machine learning technique where you take a model trained on a large dataset (like millions of images of cats, dogs, cars, etc.) and reuse it for a new but related task (like recognizing handwritten digits), even if you have only a small amount of data for your task.

How Does It Work?

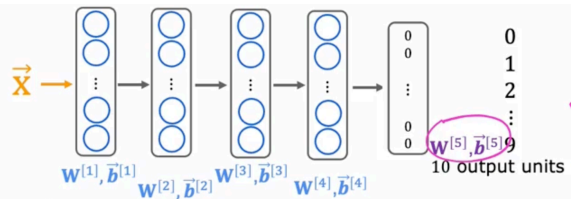
1. **Supervised Pre-training**

- Train a neural network on a **large dataset** (e.g., 1 million images, 1,000 classes).
- The model learns general features (edges, corners, shapes) in the early layers.



2. **Fine-tuning**

- **Copy** the trained model, but **replace the last layer** to match your new task (e.g., 10 output units for digits 0-9).

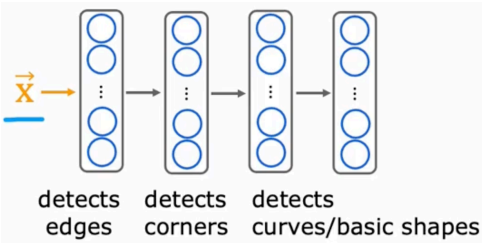


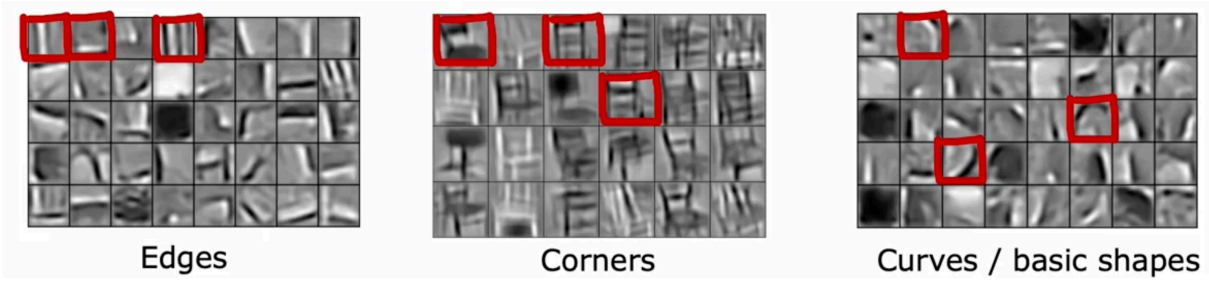
- **Train the new model** on your small dataset:
  - **Option 1:** Only train the new output layer, keep the rest fixed (only train output later parameter)
  - **Option 2:** Fine-tune all layers, starting from the pre-trained weights (training all parameters)

Why Does It Work?

- The first layers of a neural network learn to detect simple, general features like edges and shapes, which are useful for many tasks.

  (use the same input type)





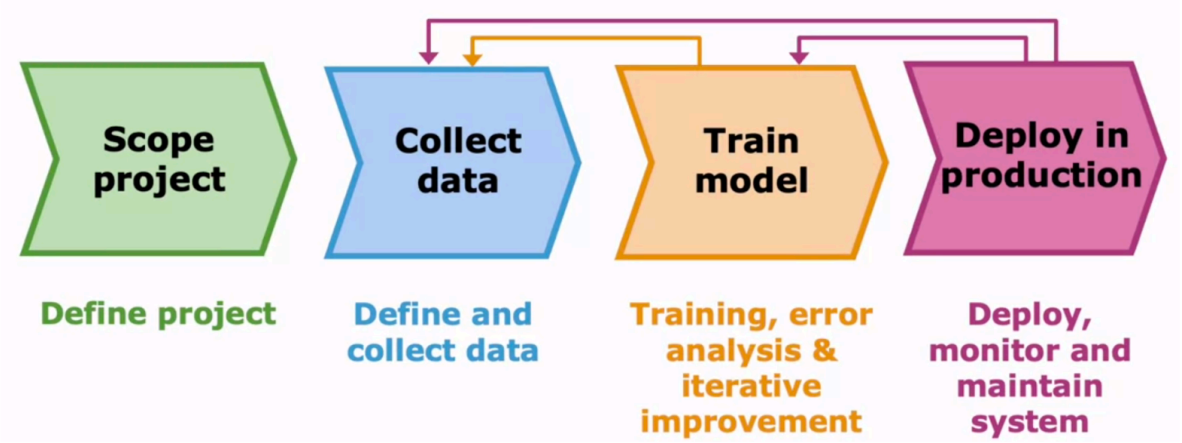
- By starting with these learned features, your new model needs much less data to work well.

Transfer learning summary

1. Download neural network parameters pretrained on a large dataset with same input type (e.g., images, audio, text) as your application (or train your own).
2. Further train (fine tune) the network on your own data.

By the way, if you've heard of advanced techniques in the news like GPT-3 or BERTs or neural networks pre-trained on ImageNet, those are actually examples of neural networks that they have someone else's pre-trained on a very large image datasets or text dataset, they can then be fine tuned on other applications.

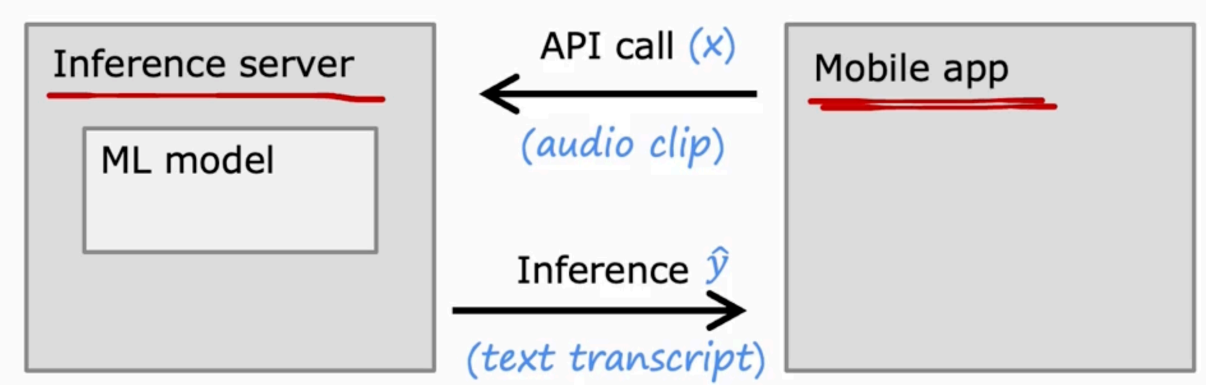
Full cycle of ML project



Deployment

After you've trained a high performing machine learning model, say a speech recognition model, a common way to deploy the model would be

- Taking your machine learning model and implement it in a server - an inference server.
    - Whose job it is to call your machine learning model, your trained model, in order to make predictions.
- If there is a user-facing application (like a mobile app), it can make API calls to the inference server to send input data and receive predictions.
    - This allows seamless interaction between the user and the machine learning model.



→ **Software engineering** may be needed for:

- Ensure reliable and efficient predictions

⇒ MLOPS: MACHINE LEARNING OPERATION

- Consider adopting MLOps practices, which focus on the

- Scaling
- Logging
- System monitoring
- Model updates

systematic deployment and maintenance of machine learning systems.

- This includes automating workflows, monitoring performance, and ensuring that the model can be updated as needed.

Fairness, bias and ethics

Ethics and Bias in Machine Learning

- Machine learning systems can significantly impact people's lives, making it crucial to ensure fairness and minimize bias.
- Historical examples, such as biased hiring tools and facial recognition systems, highlight the unacceptable consequences of neglecting these issues.

Guidelines for Ethical Development

- Assemble diverse teams to brainstorm potential harms and biases before deploying systems.
- Conduct literature searches for industry standards and guidelines to inform ethical practices.

Monitoring and Mitigation

- Audit systems for bias after training but before deployment to identify and address issues.
- Develop mitigation plans to respond to potential harms, ensuring ongoing monitoring even after deployment.

❓ **QUES:** Which of these is a way to do error analysis?

- Calculating the test error J_{test}
- Collecting additional training data in order to help the algorithm do better.
- Calculating the training error J_{train}
- Manually examine a sample of the training examples that the model misclassified in order to identify common traits and trends. 👍

Explain: Correct. By identifying similar types of errors, you can collect more data that are similar to these misclassified examples in order to train the model to improve on these types of examples

❓ **QUES:** We sometimes take an existing training example and modify it (for example, by rotating an image slightly) to create a new example with the same label. What is this process called?

- Machine learning diagnostic
- Bias/variance analysis
- Error analysis
- Data augmentation 👍

❓ **QUES:** What are two possible ways to perform transfer learning? Hint: two of the four choices are correct.

☐ Download a pre-trained model and use it for prediction without modifying or re-training it.

☑ ~~You can choose to train all parameters of the model, including the output layers, as well as the earlier layers.~~

☐ Given a dataset, pre-train and then further fine tune a neural network on the same dataset.

☑ ~~You can choose to train just the output layers' parameters and leave the other parameters of the model fixed.~~

Explain:

- Correct. It may help to train all the layers of the model on your own training set. This may take more time compared to if you just trained the parameters of the output layers.
- Correct. The earlier layers of the model may be reusable as is, because they are identifying low level features that are relevant to your task.