

# TRADE-OFFS AMONG AI TECHNIQUES


Eunsuk Kang

Required reading: Hulten, Geoff. "Building Intelligent Systems: A Guide to Machine Learning Engineering." (2018),  
Chapters 17 and 18

# LEARNING GOALS

- Organize and prioritize the relevant qualities of concern for a given project
- Explain the key ideas behind decision trees and random forests and analyze consequences for various qualities
- Explain the key ideas of deep learning and the reason for high resource needs during learning and inference and the ability for incremental learning
- Plan and execute an evaluation of the qualities of alternative AI components for a given purpose

**RECALL: ML IS A  
COMPONENT IN A SYSTEM  
IN AN ENVIRONMENT**

00:00  Offset 00:00 01:31:27

Play



Back 5s

1x

Speed



Volume

## NOTES

Write your notes here

Speaker 5 ▶ 07:44

Yeah. So there's a slight story behind that. So back when I was in, uh, Undergrad, I wrote a program for myself to measure a, the amount of time I did data entry from my father's business and I was on windows at the time and there wasn't a function called time dot [inaudible] time, uh, which I needed to parse dates to get back to time, top of representation, uh, I figured out a way to do it and I gave it to what's called the python cookbook because it just seemed like something other people could use. So it was just trying to be helpful. Uh, subsequently I had to figure out how to make it work because I didn't really have to. Basically, it bothered me that you had to input all the locale information and I figured out how to do it over the subsequent months. And actually as a graduation gift from my Undergrad, the week following, I solved it and wrote it all out.

Speaker 5 ▶ 08:38

And I asked, uh, Alex Martelli, the editor of the Python Cookbook, which had published my original recipe, a, how do I get this into python? I think it might help

How did we do on your transcript?





- Transcription model, pipeline to train the model, monitoring infrastructure
- NonML components for data storage, user interface, payment processing, ...
- User requirements and assumptions
- System quality vs model quality
- System requirements vs model requirements

# IDENTIFY RELEVANT QUALITIES OF AI COMPONENTS IN AI- ENABLED SYSTEMS

(Requirements Engineering)

AI = DL?



# ML METHODS TODAY







# ML MODEL SELECTION

HOW DO I DECIDE WHICH ML METHOD TO USE FOR MY PROJECT?

CRITERIA: QUALITY ATTRIBUTES & CONSTRAINTS

# QUALITY ATTRIBUTES

- Measurable or testable properties of a system that are used to indicate how well it satisfies its goals
- Examples
  - Performance
  - Features
  - Reliability
  - Conformance
  - Durability
  - Serviceability
  - Aesthetics
  - Perceived quality
  - and many others

Reference: Garvin, David A., [What Does Product Quality Really Mean](#). Sloan management review 25 (1984).

# ACCURACY IS NOT EVERYTHING

Beyond prediction accuracy, what qualities may be relevant for an AI component?



Speaker notes

Collect qualities on whiteboard

# QUALITIES OF INTEREST?

Scenario: AI component for transcribing audio files

the-changelog-318

[← Dashboard](#) | Quality: High ⓘ

Last saved a few seconds ago

...

Share

00:00 Offset 00:00 01:31:27

Play

Back 5s

1x

Speed

Volume

NOTES

Write your notes here

Speaker 5 ▶ 07:44

Yeah. So there's a slight story behind that. So back when I was in, **uh**, Undergrad, I wrote a program for myself to measure a, **the** amount of time I did data entry **from** my father's business and I was on windows at the time and there wasn't a function called time dot **[inaudible]** time, **uh**, which I **needed** to parse dates to get back to time, **top** of representation, **uh**, I figured out a way to do it and I gave it to what's called the python cookbook because it just seemed like something other people could use. So **it was** just trying to be helpful. **Uh**, subsequently I had to figure out how to make it work **because** I didn't really have to. Basically, it bothered me that you had to input all the **locale** information and I figured out how to do it over **the subsequent months**. And actually as a graduation gift from my Undergrad, the week following, I solved it and wrote it all out.

Speaker 5 ▶ 08:38

And I asked, **uh**, Alex Martelli, the editor of the Python Cookbook, which had published my original recipe, **a**, how do I get this into python? I think **it** might help

How did we do on your transcript? ☆☆☆☆☆



## Speaker notes

Which of the previously discussed qualities are relevant? Which additional qualities may be relevant here?

Cost per transaction; how much does it cost to transcribe? How much do we make?



# QUALITIES OF INTEREST?

Scenario: Component for detecting lane markings in a vehicle



Canny edge detection output



Hough transform output

## Speaker notes

Which of the previously discussed qualities are relevant? Which additional qualities may be relevant here?

Realtime use

# QUALITIES OF INTEREST?

Scenario: Component for detecting credit card frauds, as a service for banks



Note: Very high volume of transactions, low cost per transaction, frequent updates

# EXAMPLES OF QUALITIES TO CONSIDER

- Accuracy
- Correctness guarantees? Probabilistic guarantees (--> symbolic AI)
- How many features?
- How much data needed? Data quality important?
- Incremental training possible?
- Training time, memory need, model size -- depending on training data volume and feature size
- Inference time, energy efficiency, resources needed, scalability
- Interpretability, explainability
- Robustness, reproducibility, stability
- Security, privacy
- Fairness

# INTERPRETABILITY/EXPLAINABILITY

\*"Why did the model predict X?"\*

## Explaining predictions + Validating Models + Debugging

```
IF age between 18-20 and sex is male THEN predict arrest  
ELSE IF age between 21-23 and 2-3 prior offenses THEN predict ar  
ELSE IF more than three priors THEN predict arrest  
ELSE predict no arrest
```

Some models inherently simpler to understand

Some tools may provide post-hoc explanations

Explanations may be more or less truthful

How to measure interpretability?

**more in a later lecture**

# ROBUSTNESS



Small input modifications may change output

Small training data modifications may change predictions

How to measure robustness?

**more in a later lecture**

Image source: [OpenAI blog](#)

# FAIRNESS

*Does the model perform differently for different populations?*

```
IF age between 18-20 and sex is male THEN predict arrest  
ELSE IF age between 21-23 and 2-3 prior offenses THEN predict ar  
ELSE IF more than three priors THEN predict arrest  
ELSE predict no arrest
```

Many different notions of fairness

Often caused by bias in training data

Enforce invariants in model or apply corrections outside model

Important consideration during requirements solicitation!

**more in a later lecture**



# REVIEW: MEASURING QUALITIES

- Define a metric: Define units of interest
  - e.g., requests per second, max memory per inference, average training time in seconds for 1 million datasets
- Operationalize metric: Define measurement protocol
  - e.g., conduct experiment: train model with fixed dataset, report median training time across 5 runs, file size, average accuracy with leave-one-out cross-validation after hyperparameter tuning
  - e.g., ask 10 humans to independently label evaluation data, report reduction in error from the ML model over human predictions
- Describe all relevant factors: Inputs/experimental units used, configuration decisions and tuning, hardware used, protocol for manual steps

**On terminology:** *metric/measure* refer a method or standard format for measuring something; *operationalization* is identifying and implementing a method to measure some factor

# ON TERMINOLOGY

- Data scientists seem to speak of *model properties* when referring to accuracy, inference time, fairness, etc
  - ... but they also use this term for whether a *learning technique* can learn non-linear relationships or whether the learning algorithm is monotonic
- Software engineering wording would usually be *quality attributes* or *non-functional requirements*

# COMMON ML METHODS AND THEIR QUALITIES

# LINEAR REGRESSION: QUALITIES



- Tasks: Regression
- Qualities:
  - Advantages: ??
  - Drawbacks: ??

## Speaker notes

- Easy to interpret, low training cost, small model size
- Can't capture non-linear relationships well

# DECISION TREES

Outlook	Temperature	Humidity	Windy	Play
overcast	hot	high	false	yes
overcast	hot	high	false	no
overcast	hot	high	false	yes
overcast	cool	normal	true	yes
overcast	mild	high	true	yes
overcast	hot	normal	false	yes
rainy	mild	high	false	yes
rainy	cool	normal	false	yes
rainy	cool	normal	true	no
rainy	mild	normal	false	yes
rainy	mild	high	true	no
sunny	hot	high	false	no
sunny	hot	high	true	no
sunny	mild	high	false	no
sunny	cool	normal	false	yes
sunny	mild	normal	true	yes

$f(\text{Outlook, Temperature, Humidity, Windy}) =$



# BUILDING DECISION TREES

Outlook	Temperature	Humidity	Windy	Play
overcast	hot	high	false	yes
overcast	hot	high	false	no
overcast	hot	high	false	yes
overcast	cool	normal	true	yes
overcast	mild	high	true	yes
overcast	hot	normal	false	yes
rainy	mild	high	false	yes
rainy	cool	normal	false	yes
rainy	cool	normal	true	no
rainy	mild	normal	false	yes
rainy	mild	high	true	no
sunny	hot	high	false	no
sunny	hot	high	true	no
sunny	mild	high	false	no
sunny	cool	normal	false	yes
sunny	mild	normal	true	yes

- Identify all possible decisions
- Select the decision that best splits the dataset into distinct outcomes (typically via entropy or similar measure)
- Repeatedly further split subsets, until stopping criteria reached

# DECISION TREES: QUALITIES



- Tasks: Classification & regression
- Qualities:
  - Advantages: ??
  - Drawbacks: ??



## Speaker notes

- Easy to interpret (up to a size); can capture non-linearity; can do well with little data
- High risk of overfitting; possibly very large tree size
- Obvious ones: fairly small model size, low inference cost, no obvious incremental training; easy to interpret locally and even globally if shallow; easy to understand decision boundaries

# RANDOM FORESTS



- Train multiple trees on subsets of data or subsets of decisions.
- Return average prediction of multiple trees.
- Qualities:
  - Advantages: ??
  - Drawbacks: ??

## Speaker notes

Increased training time and model size, less prone to overfitting, more difficult to interpret

# NEURAL NETWORKS



## Speaker notes

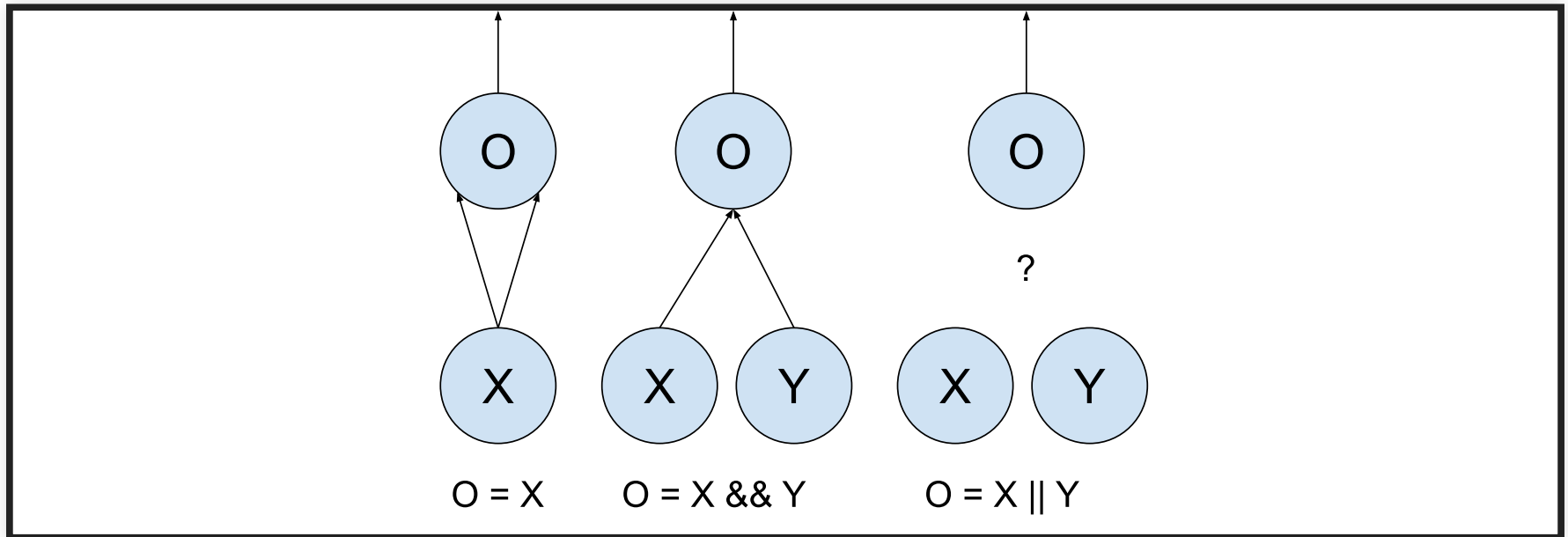
Artificial neural networks are inspired by how biological neural networks work ("groups of chemically connected or functionally associated neurons" with synapses forming connections)

From "Texture of the Nervous System of Man and the Vertebrates" by Santiago Ramón y Cajal, via [https://en.wikipedia.org/wiki/Neural\\_circuit#/media/File:Cajal\\_actx\\_inter.jpg](https://en.wikipedia.org/wiki/Neural_circuit#/media/File:Cajal_actx_inter.jpg)

# ARTIFICIAL NEURAL NETWORKS

Simulating biological neural networks of neurons (nodes) and synapses (connections), popularized in 60s and 70s

Basic building blocks: Artificial neurons, with  $n$  inputs and one output; output is activated if at least  $m$  inputs are active



(assuming at least two activated inputs needed to activate output)

# THRESHOLD LOGIC UNIT / PERCEPTRON

computing weighted sum of inputs + step function

$$z = w_1 x_1 + w_2 x_2 + \dots + w_n x_n = \mathbf{x}^T \mathbf{w}$$

e.g., step:  $\phi(z) = \begin{cases} 0 & \text{if } z < 0 \\ 1 & \text{else} \end{cases}$





$$o_1 = \phi(b_1 + w_{1,1}x_1 + w_{1,2}x_2)$$

$$o_2 = \phi(b_2 + w_{2,1}x_1 + w_{2,2}x_2)$$

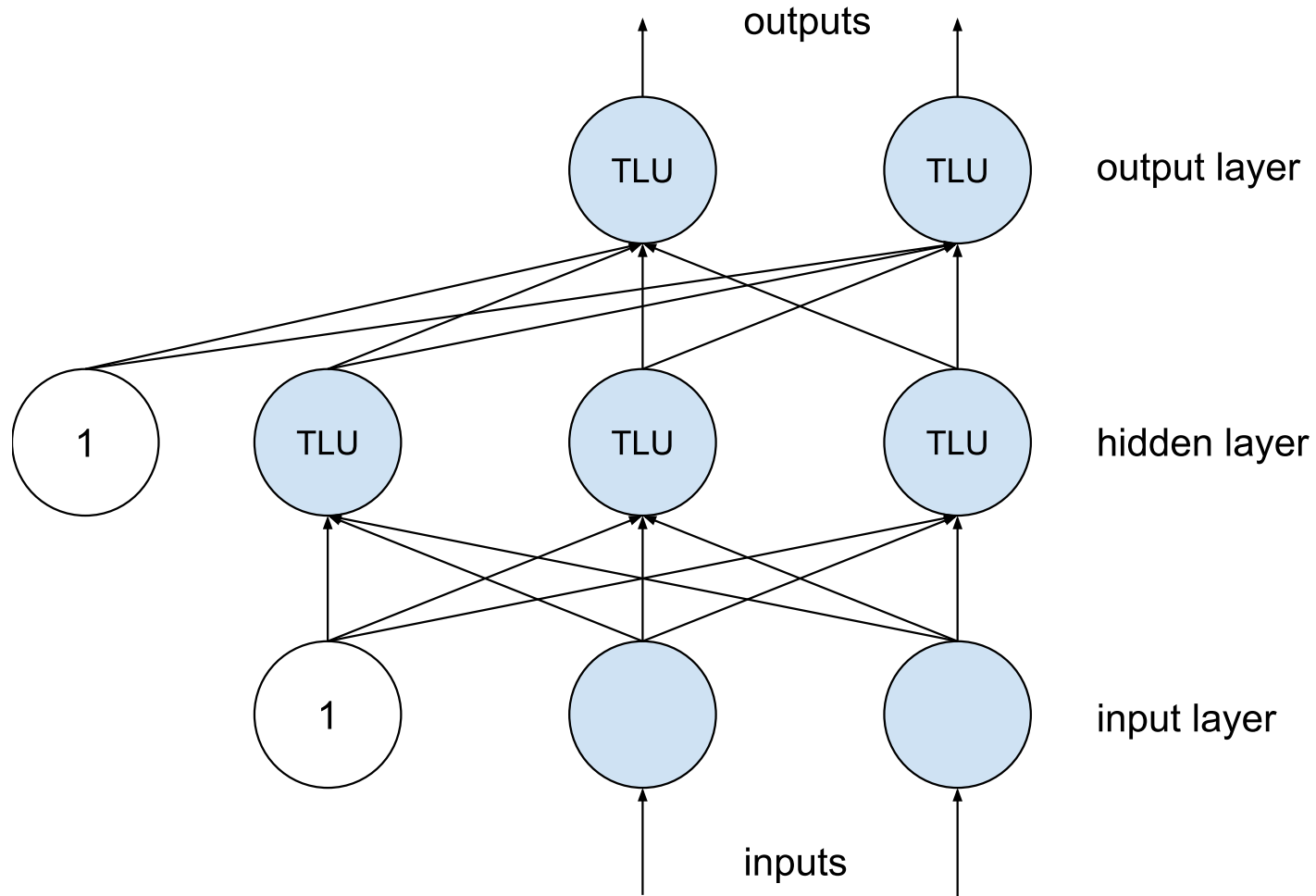
$$o_3 = \phi(b_3 + w_{3,1}x_1 + w_{3,2}x_2)$$

---

$f_{\{\mathbf{W}, \mathbf{b}\}}(\mathbf{X}) = \phi(\mathbf{W} \cdot \mathbf{X} + \mathbf{b})$   
 ( $\mathbf{W}$  and  $\mathbf{b}$  are parameters of the model)



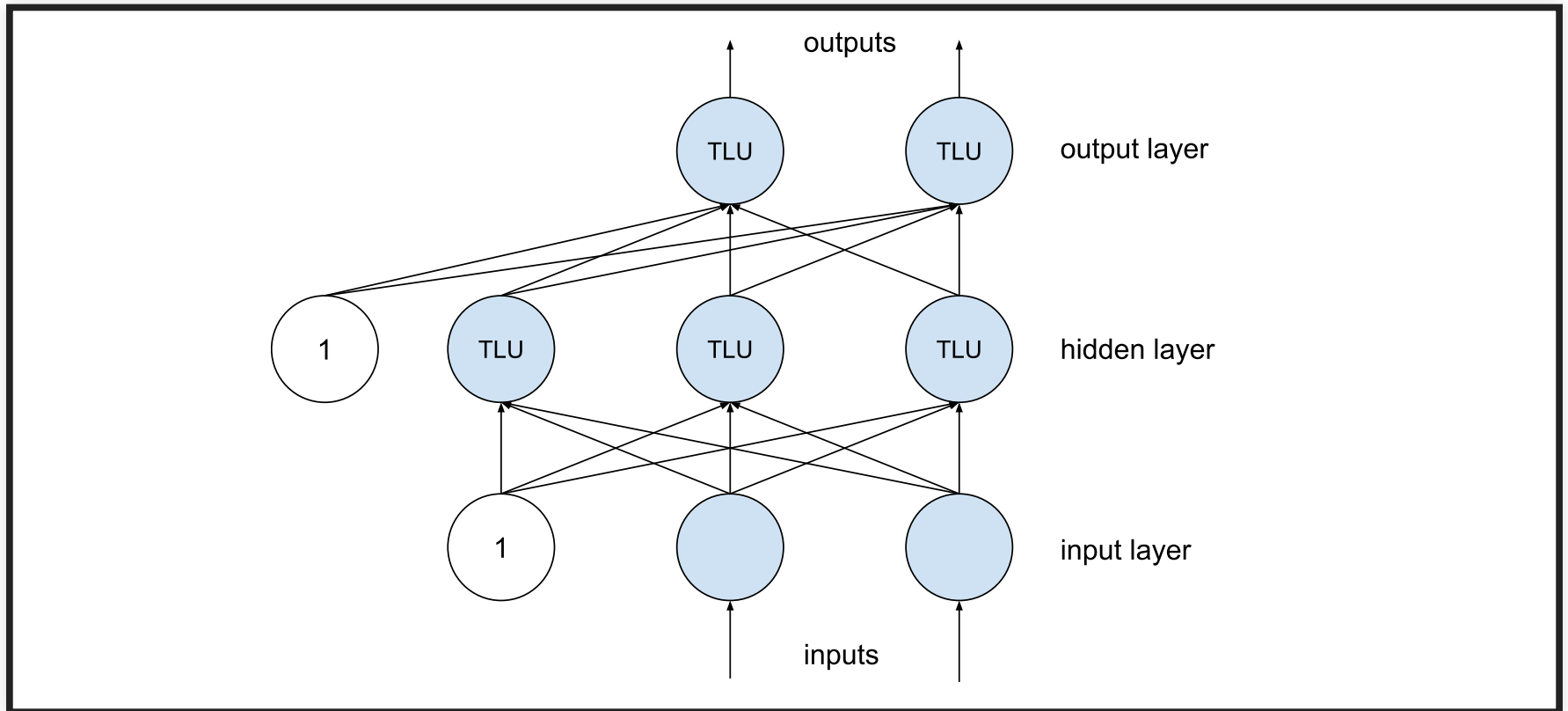
# MULTIPLE LAYERS



## Speaker notes

Layers are fully connected here, but layers may have different numbers of neurons

$$f_{\{\mathbf{W}_h, \mathbf{b}_h, \mathbf{W}_o, \mathbf{b}_o\}}(\mathbf{X}) = \phi(\mathbf{W}_o \cdot \phi(\mathbf{W}_h \cdot \mathbf{X} + \mathbf{b}_h) + \mathbf{b}_o)$$



(matrix multiplications interleaved with step function)

# LEARNING MODEL PARAMETERS (BACKPROPAGATION)

Intuition:

- Initialize all weights with random values
- Compute prediction, remembering all intermediate activations
- If predicted output has an error (measured with a loss function),
  - Compute how much each connection contributed to the error on output layer
  - Repeat computation on each lower layer
  - Tweak weights a little toward the correct output (gradient descent)
- Continue training until weights stabilize

Works efficiently only for certain  $\phi$ , typically logistic function:

$\phi(z) = 1/(1 + \exp(-z))$  or ReLU:  $\phi(z) = \max(0, z)$ .

# DEEP LEARNING

- More layers
- Layers with different numbers of neurons
- Different kinds of connections
  - Fully connected (feed forward)
  - Not fully connected (eg. convolutional networks)
  - Keeping state (eg. recurrent neural networks)
  - Skipping layers
  - ...

See Chapter 10 in Géron, Aurélien. "[Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow](#)", 2nd Edition (2019) or any other book on deep learning

## Speaker notes

Essentially the same with more layers and different kinds of architectures.

# DEEP LEARNING



- Tasks: Classification & regression
- Qualities:
  - Advantages: ??
  - Drawbacks: ??

## Speaker notes

- High accuracy; can capture a wide range of problems (linear & non-linear)
- Difficult to interpret; high training costs (time & amount of data required, hyperparameter tuning)



# EXAMPLE SCENARIO

- MNIST Fashion dataset of 70k 28x28 grayscale pixel images, 10 output classes



# EXAMPLE SCENARIO

- MNIST Fashion dataset of 70k 28x28 grayscale pixel images, 10 output classes
- 28x28 = 784 inputs in input layers (each 0..255)
- Example model with 3 layers, 300, 100, and 10 neurons

```
model = keras.models.Sequential([  
    keras.layers.Flatten(input_shape=[28, 28]),  
    keras.layers.Dense(300, activation="relu"),  
    keras.layers.Dense(100, activation="relu"),  
    keras.layers.Dense(10, activation="softmax")  
])
```

How many parameters does this model have?

# EXAMPLE SCENARIO

- MNIST Fashion dataset of 70k 28x28 grayscale pixel images, 10 output classes
- 28x28 = 784 inputs in input layers (each 0..255)
- Example model with 3 layers, 300, 100, and 10 neurons

```
model = keras.models.Sequential([
    keras.layers.Flatten(input_shape=[28, 28]),
    # 784*300+300 = 235500 parameter
    keras.layers.Dense(300, activation="relu"),
    # 300*100+100 = 30100 parameters
    keras.layers.Dense(100, activation="relu"),
    # 100*10+10 = 1010 parameters
    keras.layers.Dense(10, activation="softmax")
])
```

Total of 266,610 parameters in this small example! (Assuming float types, that's 1 MB)

# NETWORK SIZE

- 50 Layer ResNet network -- classifying 224x224 images into 1000 categories
  - 26 million weights, computes 16 million activations during inference, 168 MB to store weights as floats
- Google in 2012(!): 1TB-1PB of training data, 1 billion to 1 trillion parameters
- OpenAI's GPT-2 (2019) -- text generation
  - 48 layers, 1.5 billion weights (~12 GB to store weights)
  - released model reduced to 117 million weights
  - trained on 7-8 GPUs for 1 month with 40GB of internet text from 8 million web pages
- OpenAI's GPT-3 (2020): 96 layers, 175 billion weights, 700 GB in memory, \$4.6M in approximate compute cost for training

Speaker notes

<https://lambdalabs.com/blog/demystifying-gpt-3/>

# COST & ENERGY CONSUMPTION

Consumption	CO2 (lbs)
Air travel, 1 passenger, NY ↔ SF	1984
Human life, avg, 1 year	11,023
American life, avg, 1 year	36,156
Car, avg incl. fuel, 1 lifetime	126,000
<b>Training one model (GPU)</b>	<b>CO2 (lbs)</b>
NLP pipeline (parsing, SRL)	39
w/ tuning & experimentation	78,468
Transformer (big)	192
w/ neural architecture search	626,155

Strubell, Emma, Ananya Ganesh, and Andrew McCallum. "[Energy and Policy Considerations for Deep Learning in NLP](#)." In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp. 3645-3650. 2019.

# COST & ENERGY CONSUMPTION

Model	Hardware	Hours	CO2	Cloud cost in USD
Transformer	P100x8	84	192	289–981
ELMo	P100x3	336	262	433–1472
BERT	V100x64	79	1438	3751–13K
NAS	P100x8	274,120	626,155	943K–3.2M
GPT-2	TPUv3x32	168	—	13K–43K
GPT-3			—	4.6M

Strubell, Emma, Ananya Ganesh, and Andrew McCallum. "[Energy and Policy Considerations for Deep Learning in NLP](#)." In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp. 3645-3650. 2019.

# CONSTRAINTS AND TRADEOFFS





# ML MODEL SELECTION

HOW DO I DECIDE WHICH ML METHOD TO USE FOR MY PROJECT?

CRITERIA: QUALITY ATTRIBUTES & CONSTRAINTS

# CONSTRAINTS

Constraints define the space of attributes for valid design solutions



# TYPES OF CONSTRAINTS

- Problem constraints: Minimum required QAs for an acceptable product
- Project constraints: Deadline, project budget, available skills
- Design constraints: Type of ML task required (regression/classification), kind of available data, limits on computing resources, max. inference cost

**Q. Plausible constraints for cancer diagnosis?**



# TRADE-OFFS BETWEEN ML METHODS

- If there are multiple ML methods that satisfy the given constraints, which one do we select?
- Different ML qualities may conflict with each other; this requires making a **trade-off** between these qualities
- Among the qualities of interest, which one(s) do we care the most about?
  - And which ML method is most suitable for achieving those qualities?

# MULTI-OBJECTIVE OPTIMIZATION



- Determine optimal solutions given multiple, possibly **conflicting** objectives
- **Dominated** solution: A solution that is inferior to others in every way
- **Pareto frontier**: A set of non-dominated solutions
- Consider trade-offs among Pareto optimal solutions



# TRADE-OFFS: COST VS ACCURACY



**Netfix Prize**

Home Rules Leaderboard Update Download

## Leaderboard

Showing Test Score. [Click here to show quiz score](#)

Display top  leaders.

Rank	Team Name	Best Test Score	% Improvement	Best Submit Time
Grand Prize - RMSE = 0.8567 - Winning Team: BellKor's Pragmatic Chaos				
1	<a href="#">BellKor's Pragmatic Chaos</a>	0.8567	10.06	2009-07-26 18:18:28
2	<a href="#">The Ensemble</a>	0.8567	10.06	2009-07-26 18:38:22
3	<a href="#">Grand Prize Team</a>	0.8582	9.90	2009-07-10 21:24:40
4	<a href="#">Opera Solutions and Vandelay United</a>	0.8588	9.84	2009-07-10 01:12:31
5	<a href="#">Vandelay Industries I</a>	0.8591	9.81	2009-07-10 00:32:20
6	<a href="#">PragmaticTheory</a>	0.8594	9.77	2009-06-24 12:06:56
7	<a href="#">BellKor in BigChaos</a>	0.8601	9.70	2009-05-13 08:14:09
8	<a href="#">Dace</a>	0.8612	9.59	2009-07-24 17:18:43

*"We evaluated some of the new methods offline but the additional accuracy gains that we measured did not seem to justify the engineering effort needed to bring them into a production environment."*

Amatriain & Basilico. [Netflix Recommendations: Beyond the 5 stars](#), Netflix Technology Blog (2012)





# TRADE-OFFS: ACCURACY VS INTERPRETABILITY



Bloom & Brink. [Overcoming the Barriers to Production-Ready Machine Learning Workflows](#), Presentation at O'Reilly Strata Conference (2014).



# BREAKOUT: QUALITIES & ML METHODS



Canny edge detection output



Hough transform output

## Credit card fraud

## Lane detection

- Post answer to #lecture in Slack using template:
  - Constraints: ??
  - Qualities of interests: ??
  - ML method(s) to use: ??

# **FINDING THE RIGHT QUALITIES FOR YOUR PRODUCT**

# REQUIREMENTS ENGINEERING PROCESS

- Interview stakeholders (customers, operators, developers, business experts)
  - Understand the problem, the kind of prediction needed (e.g. classification)
  - Understand the scope: target domain, frequency of change, ...
- Broadly understand quality needs from different views
  - Model view: direct expectation on the model(s)
  - Data view: availability, quantity, and quality of data
  - System view: understand system goals and role of ML model and interactions with environment
  - Infrastructure view: training cost, reproducibility needs, serving infrastructure needs, monitoring needs, ...
  - Environment/user view: external expectations on the system by users and society, e.g. fairness, safety
- Collect and document needs, resolve conflicts, discuss and prioritize

Siebert, Julien, Lisa Joeckel, Jens Heidrich, Koji Nakamichi, Kyoko Ohashi, Isao Namba, Rieko Yamamoto, and Mikio Aoyama. "[Towards Guidelines for Assessing Qualities of Machine Learning Systems](#)." In International Conference on the Quality of Information and Communications Technology, pp. 17-31. Springer, Cham, 2020.



# REQUIREMENTS ENGINEERING PROCESS

- Set minimum accuracy expectations ("functional requirement")
- Identify runtime needs (how many predictions, latency requirements, cost budget, mobile vs cloud deployment)
- Identify evolution needs (update and retrain frequency, ...)
- Identify explainability needs
- Identify protected characteristics and possible fairness concerns
- Identify security and privacy requirements (ethical and legal), e.g., possible use of data
- Understand data availability and need (quality, quantity, diversity, formats, provenance)
- Involve data scientists and legal experts
- **Map system goals to AI components**

Further reading: Vogelsang, Andreas, and Markus Borg. "[Requirements Engineering for Machine Learning: Perspectives from Data Scientists](#)." In Proc. of the 6th International Workshop on Artificial Intelligence for Requirements Engineering (AIRE), 2019.

# SUMMARY

- Quality is multifaceted
- Requirements engineering to solicit important qualities and constraints
- Many qualities of interest, define metrics and operationalize
- Constraints and tradeoff analysis for selecting ML techniques in production ML settings