Week 1 discussion

MACHINE LEARNING

Dr. Temitayo Olugbade

UNIVERSITY
OF SUSSEX

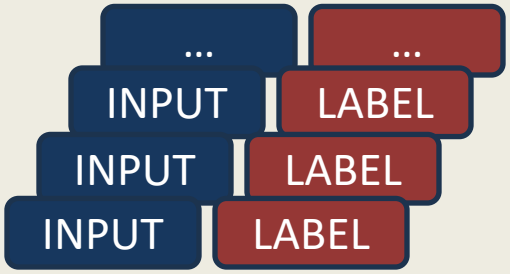
Week 1 mini-video content

This week, you've been looking at:

- What is machine learning?
- A simple machine learning model

Recall from Week 1 mini-videos

Event A, Response A
Event B, Response B
Event C, Response C
Event D, Response D
Event E, Response E
…



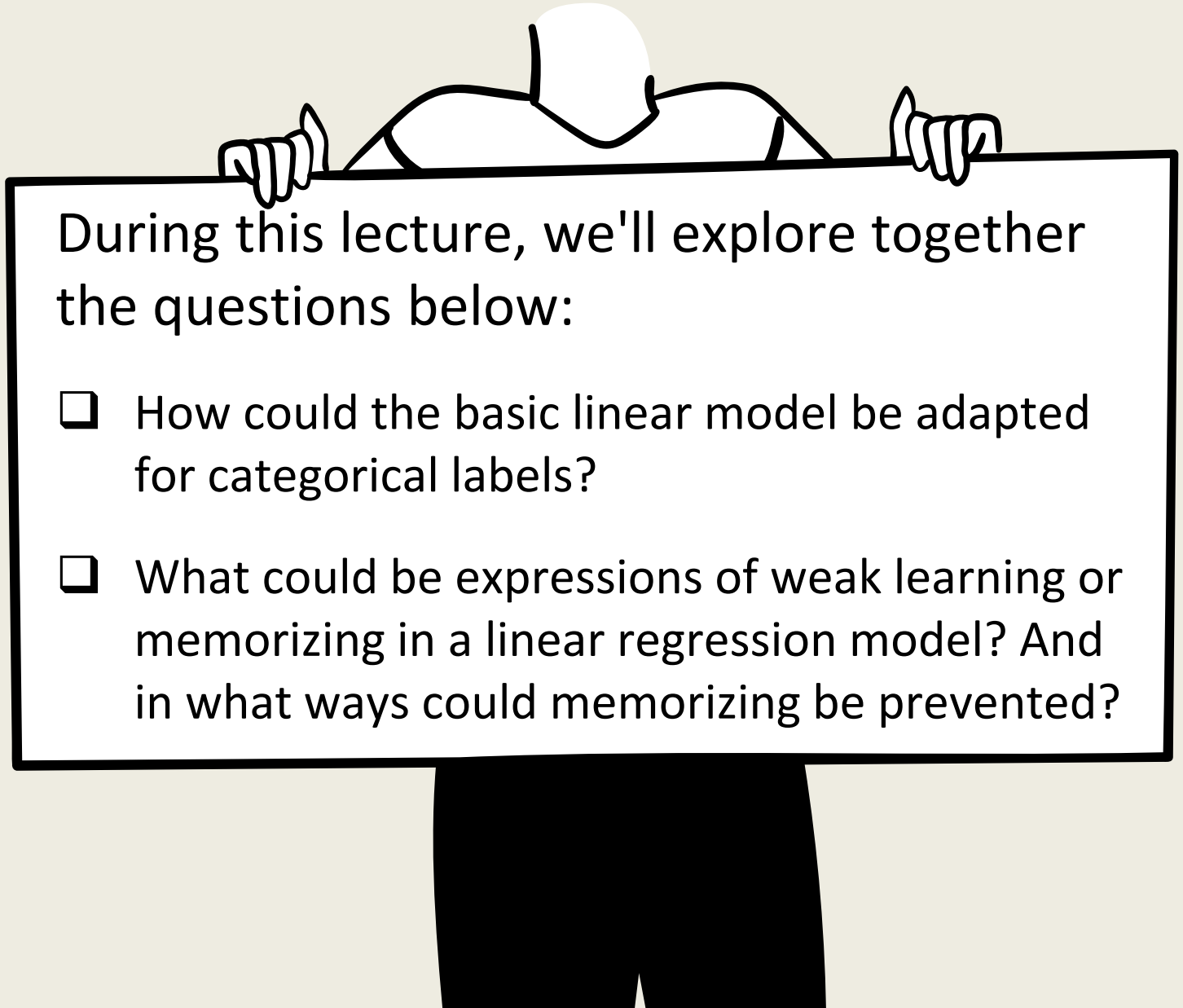
Consider that there's a process you could automate

…and a rule-based approach would not work

You would need:

- To train a ML model
- Training data instances in (input/features, output/label) pairs
    - The model output should be exactly or very close to the true label in the training data
    - The same should hold for instances not included in the training data – **generalizability**

Learning outcomes



During this lecture, we'll explore together the questions below:

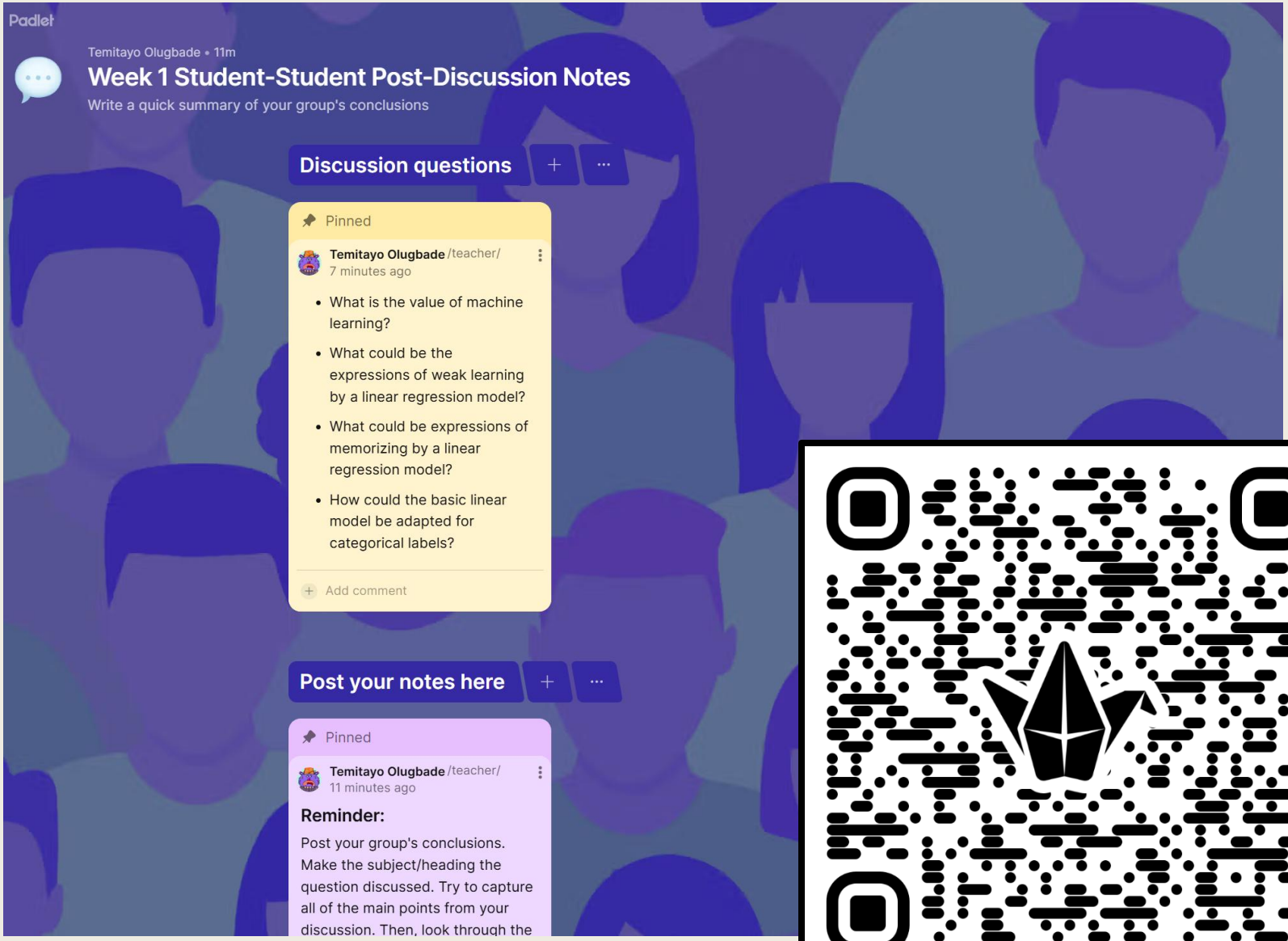
- ❑ How could the basic linear model be adapted for categorical labels?
- ❑ What could be expressions of weak learning or memorizing in a linear regression model? And in what ways could memorizing be prevented?

Discussion outline

- How could the basic linear model be adapted for categorical labels? (35 mins)

- What could be expressions of weak learning or memorizing in a linear regression model? And in what ways could memorizing be prevented? (35 mins)

Discuss & Note down summary



Discussion outline

- **How could the basic linear model be adapted for categorical labels?** (35 mins)

- What could be expressions of weak learning or memorizing in a linear regression model? And in what ways could memorizing be prevented? (35 mins)

Toy data with categorical labels

labels

$\{y_n\}_{n=1}^{6}, D_y = 1$

features

$\{\boldsymbol{x}_n\}_{n=1}^{6}, D_x = (height, width, channel)$

$channel = 3$ for R,G,B

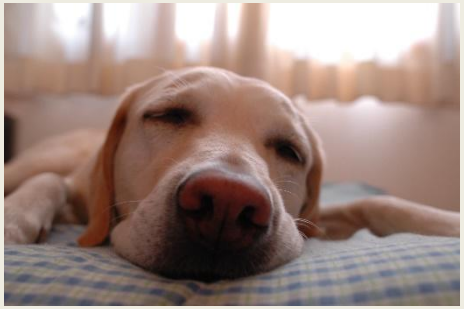
cat

Source: Muhammad Mahdi Karim
https://commons.wikimedia.org/wiki/File:Domestic_cat_felis_catus.jpg

Source: Dimitri Torterat
https://commons.wikimedia.org/wiki/File:Domestic_shorthaired_cat_face.jpg

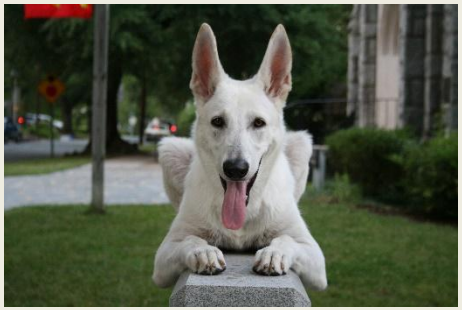
Source: Peter Forster
https://commons.wikimedia.org/wiki/File:Cat_Briciola_with_pretty_and_different_colour_of_eyes.jpg

dog



Source: Eugene0126jp
https://en.wikipedia.org/wiki/File:Dog.in.sleep.jpg

Source: Jina Lee
https://commons.wikimedia.org/wiki/File:Pug_dog_nose_face_detail.JPG



Source: IldarSagdejev
https://en.wikipedia.org/wiki/File:2008-06-26_White_German_Shepherd_Dog_Posing_3.jpg

The basic linear model & Classification

Basic linear ML model - $f(\boldsymbol{x}) = \hat{\boldsymbol{y}} = \boldsymbol{xw} + b$

Class question – Why do we say that it's a regression model, not classification?

Student-student discussion – How could the basic linear model be adapted for categorical labels? ***(10mins)***

Hint – What could be done to $\boldsymbol{xw} + b$ to make it discrete/categorical?

& what loss function would be helpful here? ***(10mins)***

Hint – Think about the nature of the true label in classification tasks.

Basic linear model for classification

$$f(\boldsymbol{x}) = \sigma(\boldsymbol{x}\boldsymbol{w} + b) = \hat{\boldsymbol{y}}$$

where

- $f(\cdot)$ – basic linear model
- $\boldsymbol{x}$ – features (or model input)
- $\hat{\boldsymbol{y}}$ – predicted labels/targets (or model output)
- $\boldsymbol{w}$, b – weights, bias (or model parameters)
- $\sigma(\cdot)$ – activation function (for discretizing real values)

Toy data with numerical categorical labels

labels

$\{y_n\}_{n=1}^{6}, D_y = 1$

+1

-1

features

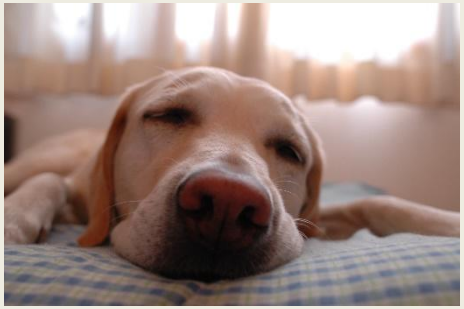
$\{\boldsymbol{x}_n\}_{n=1}^{6}, D_x = (height, width, channel)$

$channel = 3$ for R,G,B

Source: Muhammad Mahdi Karim
https://commons.wikimedia.org/wiki/File:Domestic_cat_felis_catus.jpg

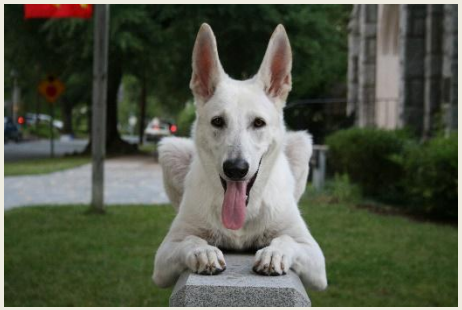
Source: Dimitri Torterat
https://commons.wikimedia.org/wiki/File:Domestic_shorthaired_cat_face.jpg

Source: Peter Forster
https://commons.wikimedia.org/wiki/File:Cat_Briciola_with_pretty_and_different_colour_of_eyes.jpg



Source: Eugene0126jp
https://en.wikipedia.org/wiki/File:Dog.in.sleep.jpg

Source: Jina Lee
https://commons.wikimedia.org/wiki/File:Pug_dog_nose_face_detail.JPG



Source: IldarSagdejev
https://en.wikipedia.org/wiki/File:2008-06-26_White_German_Shepherd_Dog_Posing_3.jpg

Sign activation function

$$f(\boldsymbol{x}) = \sigma(\boldsymbol{xw} + b) = \hat{\boldsymbol{y}}$$

- Sign activation function could be used to make real/continuous-valued $\boldsymbol{xw} + b$ always fall into one of classes -1 or $+1$

- If $\boldsymbol{xw} + b > 0$, $\hat{y} = +1$, i.e. class $+1$
  If $\boldsymbol{xw} + b < 0$, $\hat{y} = -1$, i.e. class -1

Potential loss function

- Sign loss

$$L_0 = \frac{1}{N}\sum_{n=1}^{N} \boldsymbol{I}.(sign(\boldsymbol{x}_n\boldsymbol{w} + b) \neq y_n)$$

Potential loss function

- Sign loss

does the sign of $\boldsymbol{xw} + b$ NOT match the sign of the true class?

$$L_0 = \frac{1}{N}\sum_{n=1}^{N} \boldsymbol{I}.(sign(\boldsymbol{x}_n\boldsymbol{w} + b) \neq y_n)$$

Potential loss function

- Sign loss

$$L_0 = \frac{1}{N}\sum_{n=1}^{N} \boldsymbol{I}.(sign(\boldsymbol{x}_n\boldsymbol{w} + b) \neq y_n)$$

1

Potential loss function

- Sign loss **(aka 0–1 loss)**

$$L_0 = \frac{1}{N}\sum_{n=1}^{N} \boldsymbol{I}.(sign(\boldsymbol{x}_n\boldsymbol{w} + b) \neq y_n)$$

- Hinge loss

$$L_{hinge} = \frac{1}{N}\sum_{n=1}^{N} max\big(0, \quad -\boldsymbol{y}_n(\boldsymbol{x}_n\boldsymbol{w} + b)\big)$$

Do $\boldsymbol{y}$ (true class) and $\boldsymbol{xw}$ (prediction) have the same sign?

✓ Then $-\boldsymbol{y}_n(\boldsymbol{x}_n\boldsymbol{w} + b)$ will negative and 0 loss is recorded

x Then $-\boldsymbol{y}_n(\boldsymbol{x}_n\boldsymbol{w} + b)$ will positive and $-\boldsymbol{y}_n(\boldsymbol{x}_n\boldsymbol{w} + b)$ is recorded

Discussion outline

- How could the basic linear model be adapted for categorical labels? (35 mins)

- **What could be expressions of weak learning or memorizing in a linear regression model? And in what ways could memorizing be prevented?** (35 mins)

Basic linear model & ML goal

Goal – To find a linear model such that:

A. its output is exactly or very close to the true label for instances in the training data, and

B. it is generalizable, i.e. its output is exactly or very close to the true label for instances NOT in the training data

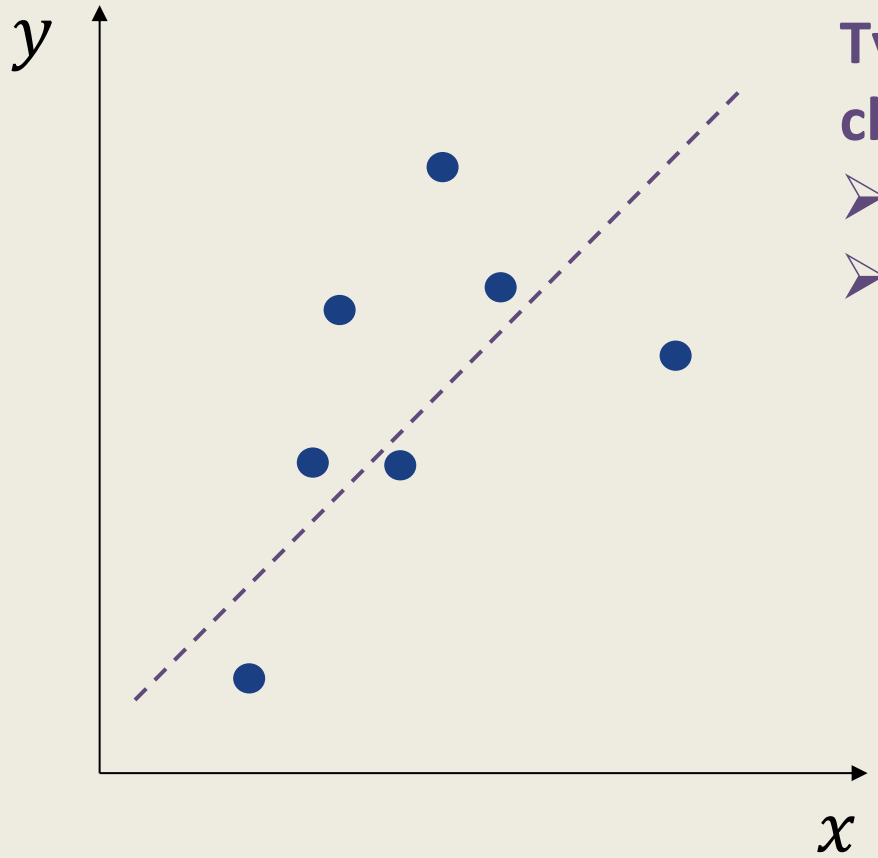
Student-student discussion *(10mins)*

- What does failing at A imply (about the ML model)?
- What does failing at B imply (about the ML model)?

Weak learner

Failing at Goal A → weak learner

i.e. $\hat{y}_n$ (model output) not close to y_n (true label) for the training data



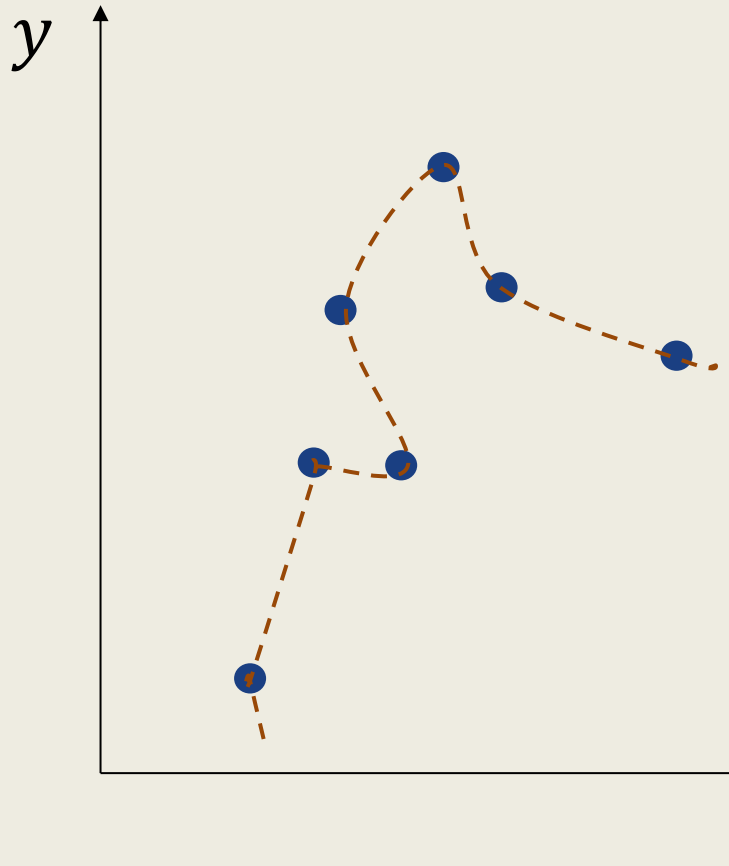
Typical weak learner characteristics

- Low model complexity
- High bias (strong assumption)

Training data memorization

Failing at Goal B but not A → learner has memorized the training data

i.e. the model performs well for training data, but not other data



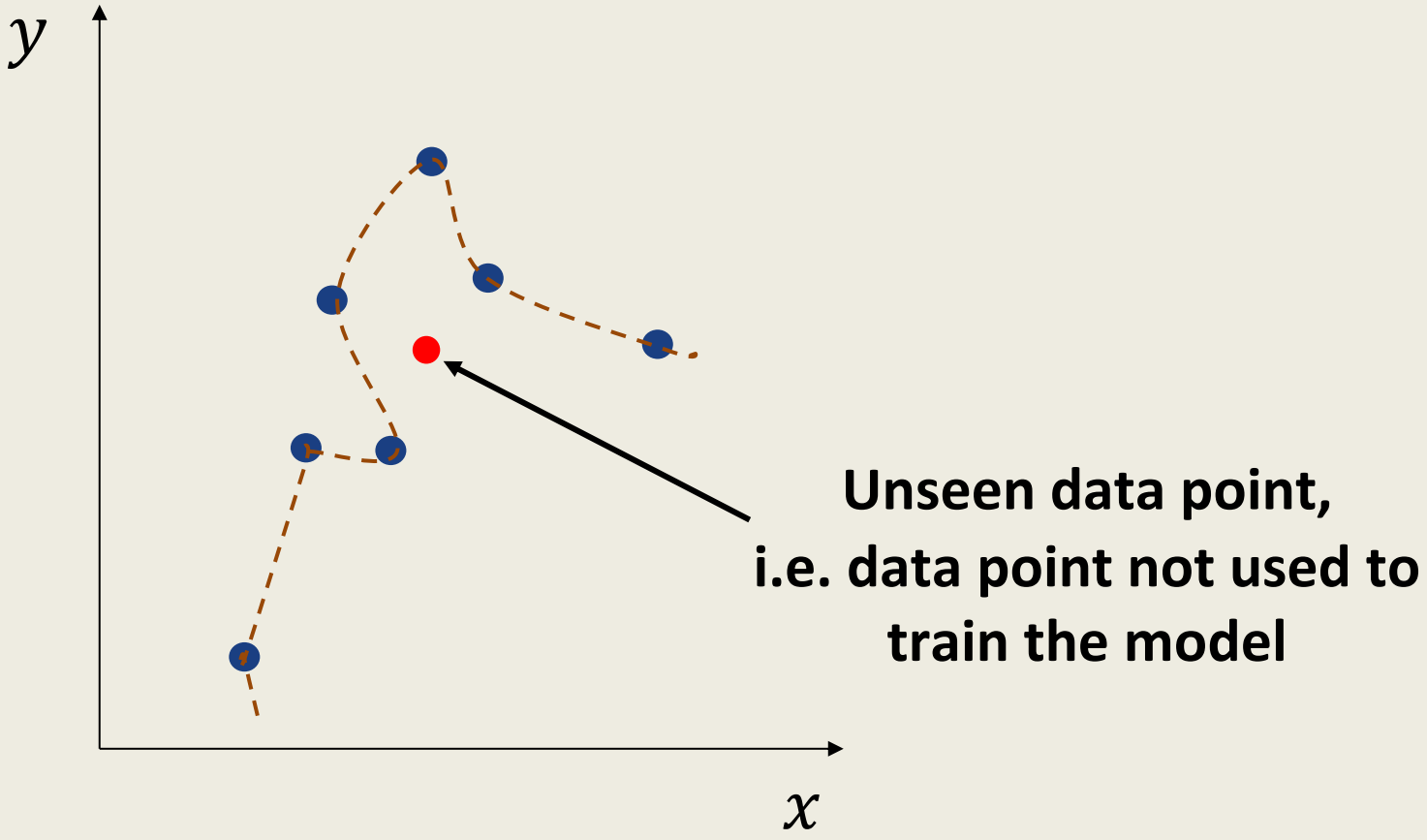
Typical memorization characteristics

- High model complexity
- High variance (strong sensitivity to seen data)

Overfitting to the training data

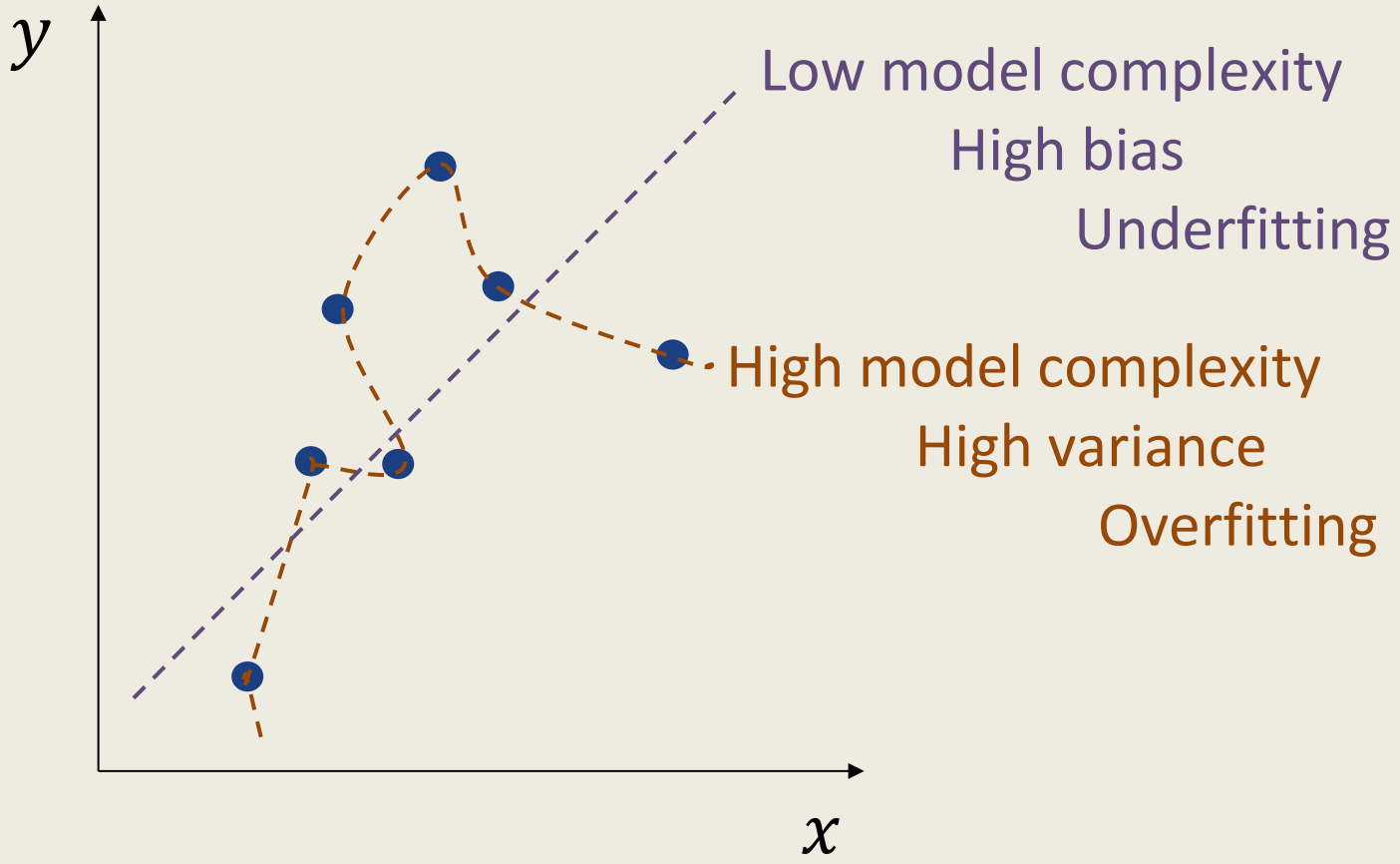
Failing at Goal B but not A

i.e. the model performs well for training data, but not other data



Generalizability in ML

Sweet spot between weak learning and memorization



Addressing overfitting

Student-student discussion *(10mins)*

– How could the problem of overfitting be addressed for the basic linear model?

Model weights & Model complexity

$$f(\boldsymbol{x}) = \boldsymbol{x}\boldsymbol{w} + b = \; b + w_1 x_1 + w_2 x_2 + \cdots w_D x_D$$

- If some weights are zero

$$\text{e.g. } f(\boldsymbol{x}) = b + 0x_1 + w_2 x_2 + 0x_3$$

→ the model is then less complex

If some weights are zero while the model still minimizes prediction loss

→ uninformative features are being ignored

i.e. the model is learning which features are informative and weigh them in accordingly

Encouraging lower complexity

L1 regularization (aka lasso regression)

- Add L1 penalty to the loss function, i.e.

$$L_{lasso} = \frac{1}{N}\sum_{n=1}^{N}(\boldsymbol{x}_n\boldsymbol{w} + b - \boldsymbol{y}_n)^2 + \alpha \cdot |w|$$

where

- $\alpha|w|$ – regularization term
- w – weights (model parameter)
- α – regularization strength (model hyperparameter)

Encouraging lower complexity

L1 regularization

- Add an L1 penalty to the loss function, i.e.

$$L_{lasso} = \frac{1}{N}\sum_{n=1}^{N}(\boldsymbol{x}_n\boldsymbol{w} + b - \boldsymbol{y}_n)^2 + \alpha \cdot |w|$$

loss function
mean squared error,
i.e. L2 loss

where

- $\alpha|w|$ – regularization term
- w – weights (model parameter)
- α – regularization strength (model hyperparameter)

Encouraging lower complexity

L1 regularization

- Add an L1 penalty to the loss function, i.e.

$$L_{lasso} = \frac{1}{N}\sum_{n=1}^{N}(\boldsymbol{x}_n\boldsymbol{w} + b - \boldsymbol{y}_n)^2 + \alpha \cdot |w|$$

absolute value → L1

where

- $\alpha|w|$ – regularization term
- w – weights (model parameter)
- α – regularization strength (model hyperparameter)

Encouraging lower complexity

L1 regularization

- Add an L1 penalty to the loss function, i.e.

$$L_{lasso} = \frac{1}{N}\sum_{n=1}^{N}(\boldsymbol{x}_n\boldsymbol{w} + b - \boldsymbol{y}_n)^2 + \alpha \cdot |w|$$

where

  - $\alpha|w|$ – regularization term
  - w – weights (model parameter)
  - α – regularization strength (model hyperparameter)

- This penalizes non-zero weights
  - i.e. it encourages zero weights
  - zero weights imply reduced model complexity

Encouraging lower complexity – Alternative

L2 regularization (aka ridge regression)

- Add an L2 penalty to the loss function, i.e.

$$L_{ridge} = \frac{1}{N}\sum_{n=1}^{N}(\boldsymbol{x}_n\boldsymbol{w} + b - \boldsymbol{y}_n)^2 + \alpha\|\boldsymbol{w}\|^2$$

Encouraging lower complexity – Alternative

L2 regularization

- Add an L2 penalty to the loss function, i.e.

$$L_{lasso} = \frac{1}{N}\sum_{n=1}^{N}(\boldsymbol{x}_n\boldsymbol{w} + b - \boldsymbol{y}_n)^2 + \alpha\|\boldsymbol{w}\|^2$$

squared value → L2

Encouraging lower complexity – Alternative

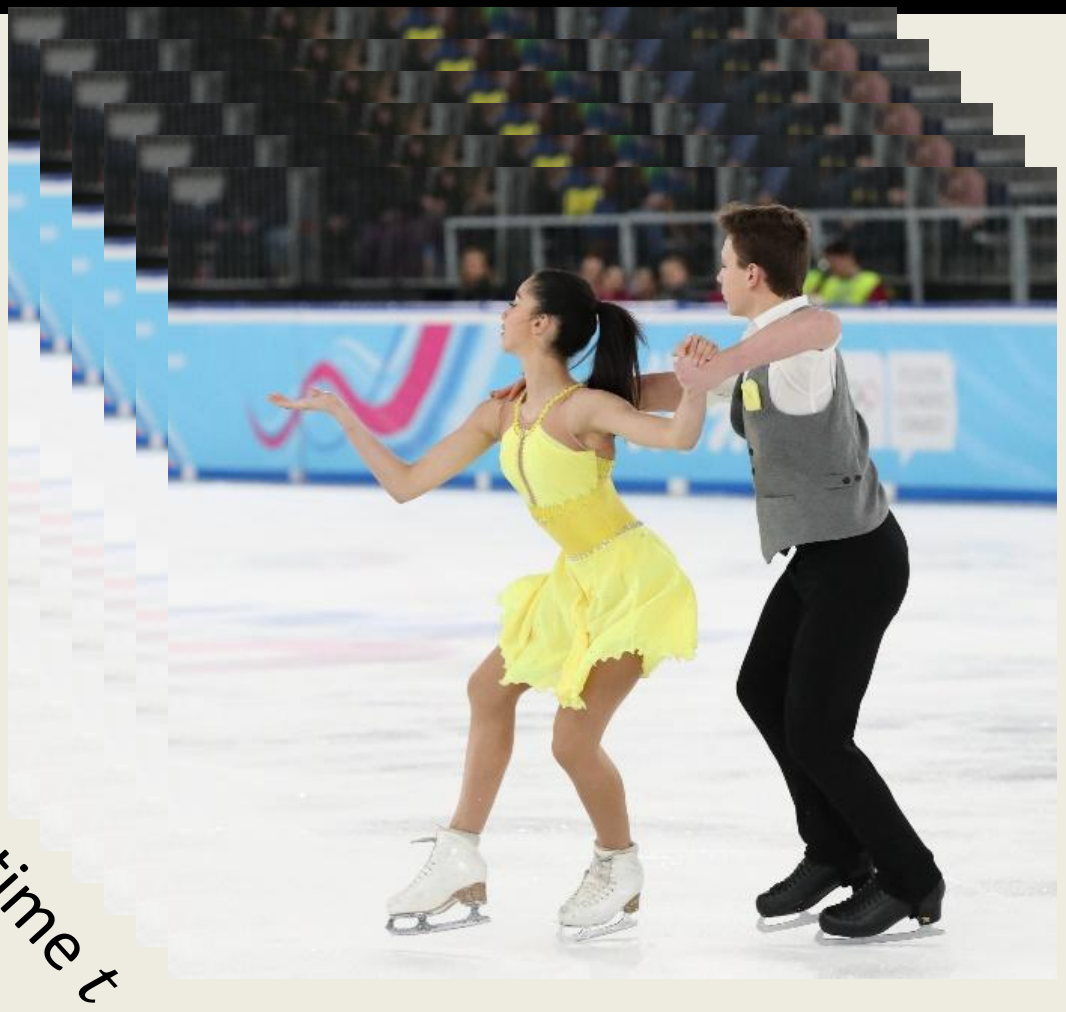
L2 regularization

- Add an L1 penalty to the loss function, i.e.

$$L_{lasso} = \frac{1}{N}\sum_{n=1}^{N}(\boldsymbol{x}_n\boldsymbol{w} + b - \boldsymbol{y}_n)^2 + \alpha\|\boldsymbol{w}\|^2$$

- This penalizes large weights
  - i.e. it encourages very small weights
  - much smaller weights reduce model complexity

Toy video data

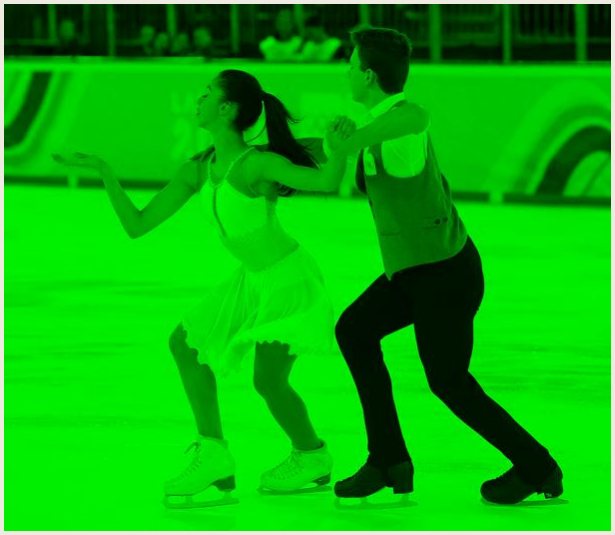


Video data dimensionality $t \times h \times w \times c$

Source: Sandro Halank
https://commons.wikimedia.org/wiki/File:2020-01-11_Ice_Dance_Rhythm_Dance_%282020_Winter_Youth_Olympics%29_by_Sandro_Halank%E2%80%930857.jpg

Toy video data: Channels

Red channel c_1



Green channel c_2

Blue channel c_3

Video data dimensionality $t \times h \times w \times c$

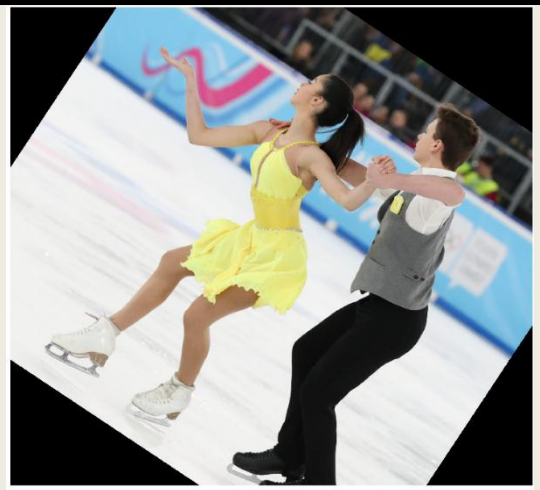
Class question

?

What components of this image make it relevant to the label '*skating*'?

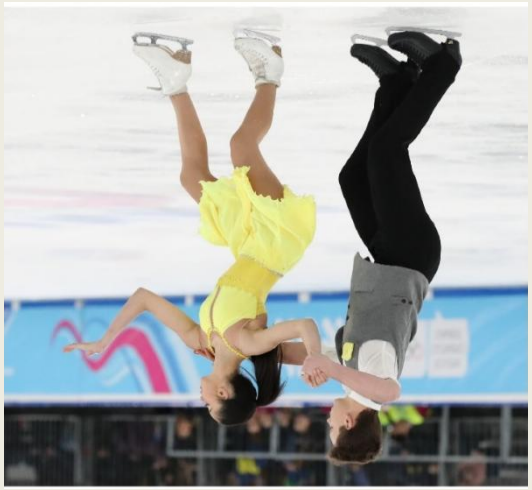
How could you encourage a ML model to learn all the different components of the image that are relevant to the label, rather than simple memorizing the image and its label?

Transformations in $h \times w$ dimension

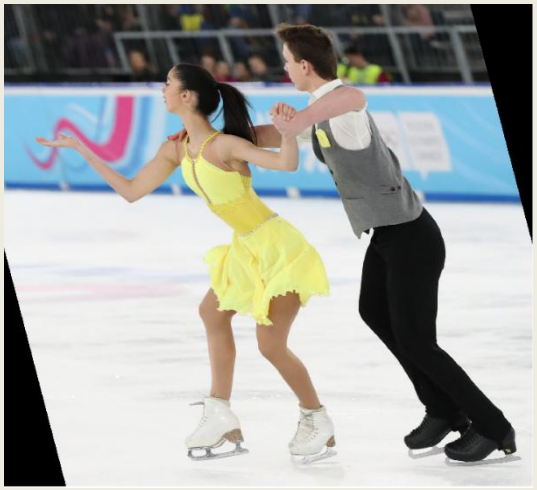


Rotation randomly, e.g. between [-45°, 45°]

Translation randomly, e.g. between [-250, 250]



Flipping (reflection) randomly



Horizontal shear transformation randomly, e.g. [-30°, 30°]

Cropping randomly

Transformations in channel dimension

Colour transformation randomly

Adding noise (Gaussian) randomly

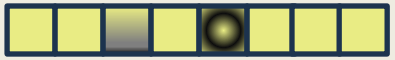
Blurring randomly

Contrast transformation randomly

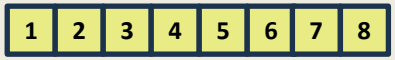
Transformation to greyscale randomly

Transformations in time dimension

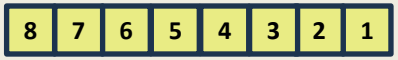
- Jittering – adding noise randomly along the time dimension



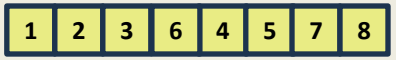
- Time warping – randomly changing the sampling rates of random subsegments along the time dimension



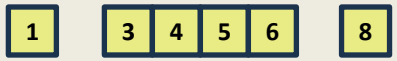
- Time reversal – flipping temporal order



- Jigsaw – randomly shuffling subsegments along the time dimension



- Dropout – randomly masking time steps in the signal



Addressing overfitting

- **Student-student discussion** – How could the problem of overfitting be addressed for the basic linear model?

  ✓ Encourage lower model complexity

  Regularization

  ✓ Encourage learning of informative features

  Data augmentation

Summary

1. An activation function allows the basic linear model to be used for classification.
2. Classification loss functions are typically different from regression loss functions, although differentiability is important for both.
3. Achieving the goal of generalizability to unseen data is trade-off between **bias** (**underfitting**) & **variance** (**overfitting**).
4. Overfitting can be addressed with **regularization** or **data augmentation**.