

CS 188: Artificial Intelligence

Machine Learning



Summer 2024: Eve Fleisig & Evgeny Pobachienko

Demo: Catching AI-Generated Text

- Feature design
 - Complexity in feature design vs. model design
- Evaluation: accuracy, precision & recall, F1 score
- Generalization, calibration, robustness

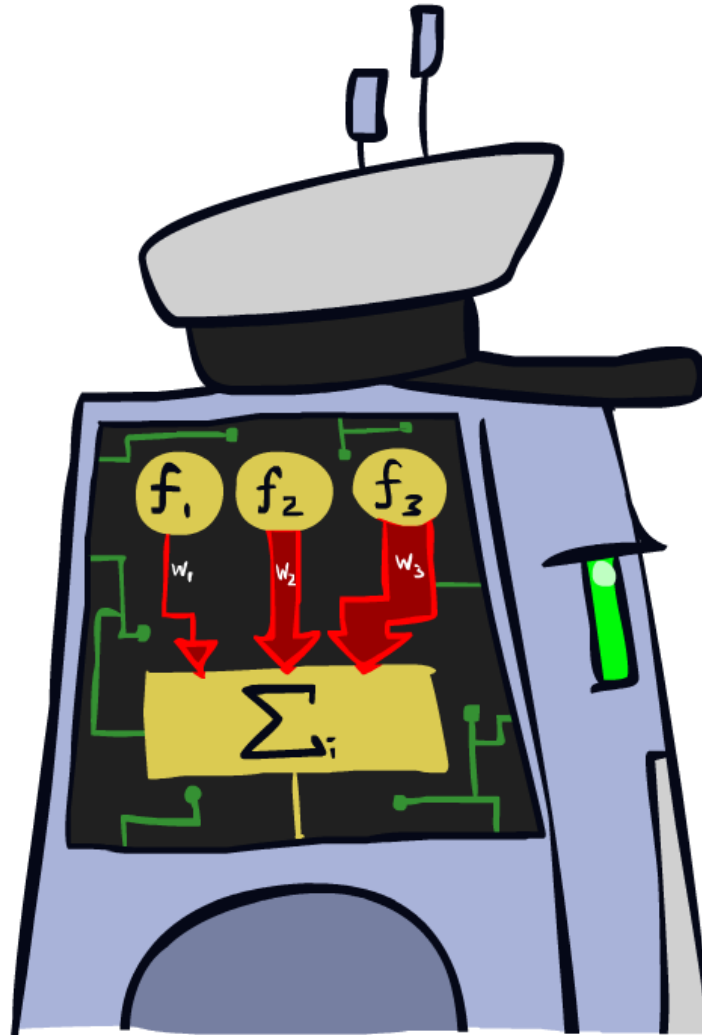
$$\text{Precision} = \frac{TP}{TP + FP} \quad \text{Recall} = \frac{TP}{TP + FN}$$

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + FN + TN}$$

$$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

		True Class	
		Positive	Negative
Predicted Class	Positive	TP	FP
	Negative	FN	TN

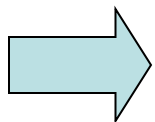
Linear Classifiers



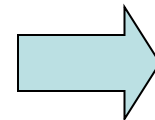
Feature Vectors

 x $f(x)$ y

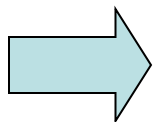
```
Hello,  
  
Do you want free printer  
cartridges? Why pay more  
when you can get them  
ABSOLUTELY FREE! Just
```



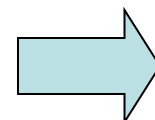
```
# free      : 2  
YOUR_NAME   : 0  
MISPELLED   : 2  
FROM_FRIEND : 0  
...
```



SPAM
or
+



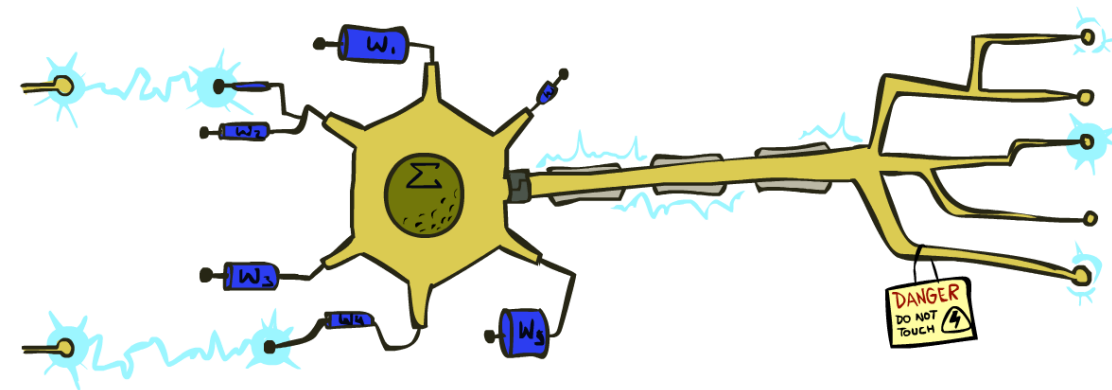
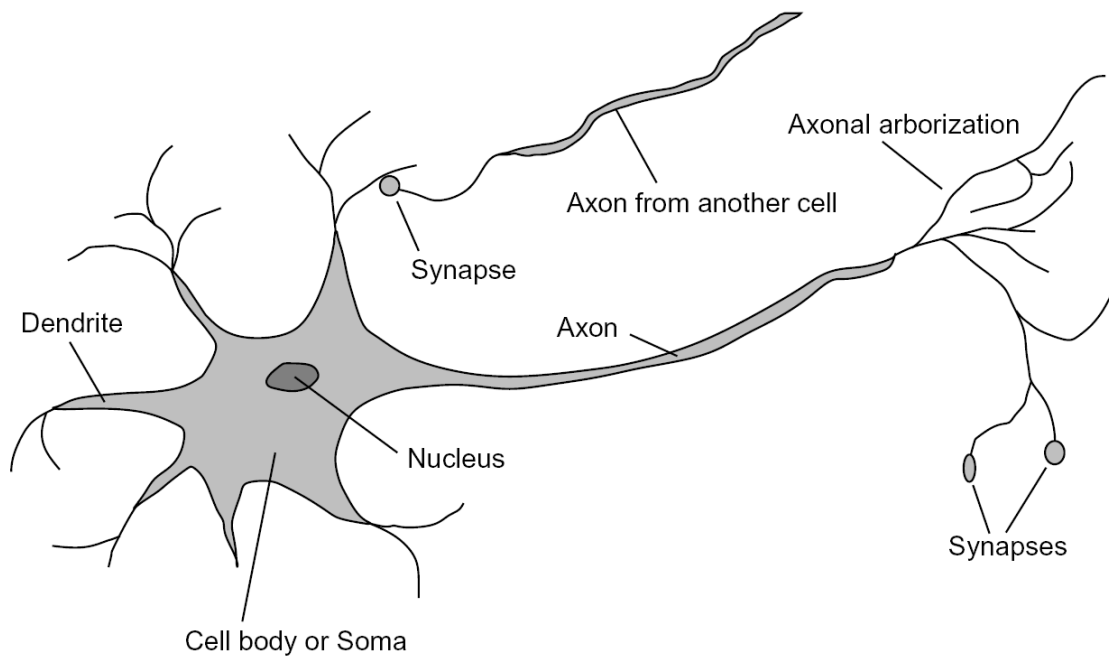
```
PIXEL-7,12 : 1  
PIXEL-7,13 : 0  
...  
NUM_LOOPS  : 1  
...
```



"2"

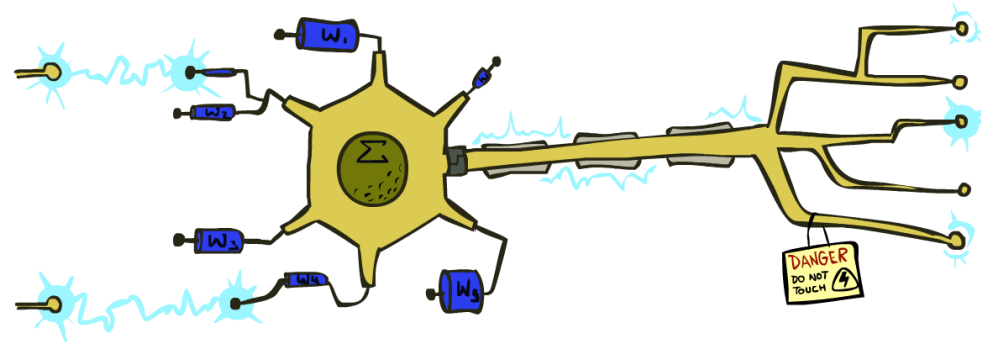
Some (Simplified) Biology

- Very loose inspiration: human neurons



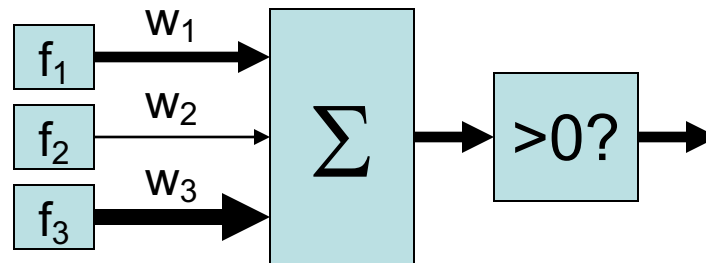
Linear Classifiers

- Inputs are **feature values**
- Each feature has a **weight**
- Sum is the **activation**



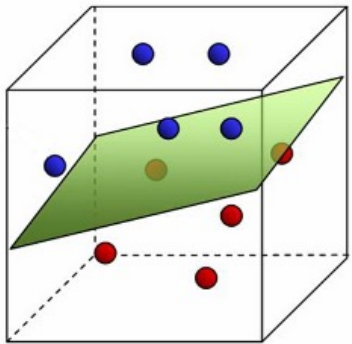
$$\text{activation}_w(x) = \sum_i w_i \cdot f_i(x) = w \cdot f(x)$$

- If the activation is:
 - Positive, output +1
 - Negative, output -1



Weights

- Binary case: compare features to a weight vector
- Learning: figure out the weight vector from examples



```
(  
  # free      : 4  
  YOUR_NAME   :-1  
  MISPELLED   : 1  
  FROM_FRIEND :-3  
  ...  
)
```

w

$f(x_1)$

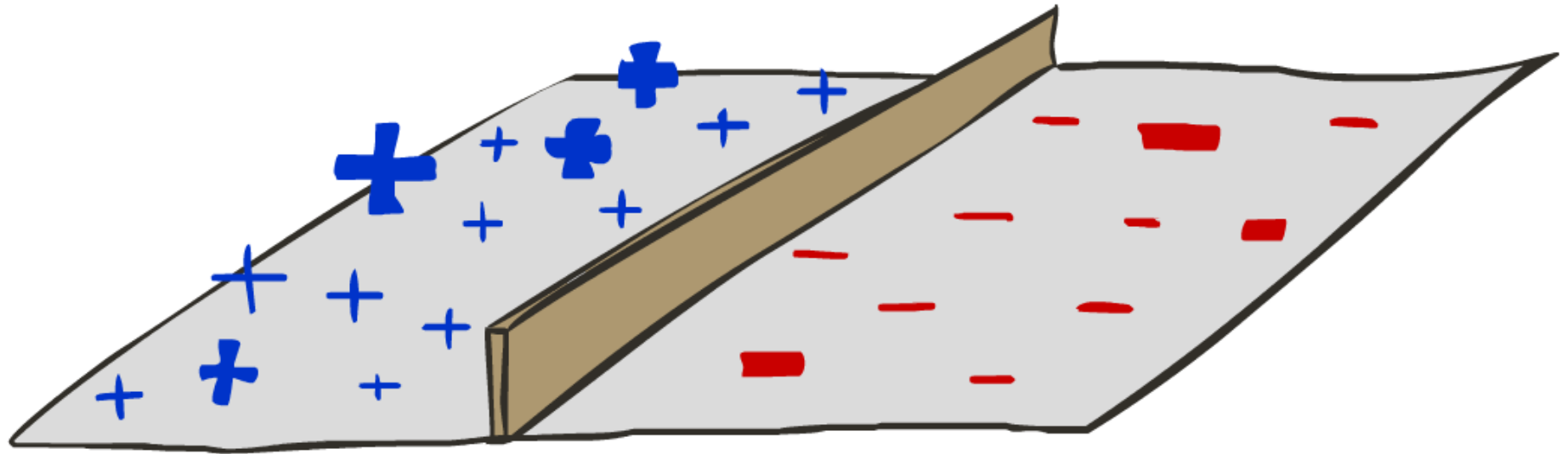
```
(  
  # free      : 2  
  YOUR_NAME   : 0  
  MISPELLED   : 2  
  FROM_FRIEND : 0  
  ...  
)
```

$f(x_2)$

```
(  
  # free      : 0  
  YOUR_NAME   : 1  
  MISPELLED   : 1  
  FROM_FRIEND : 1  
  ...  
)
```

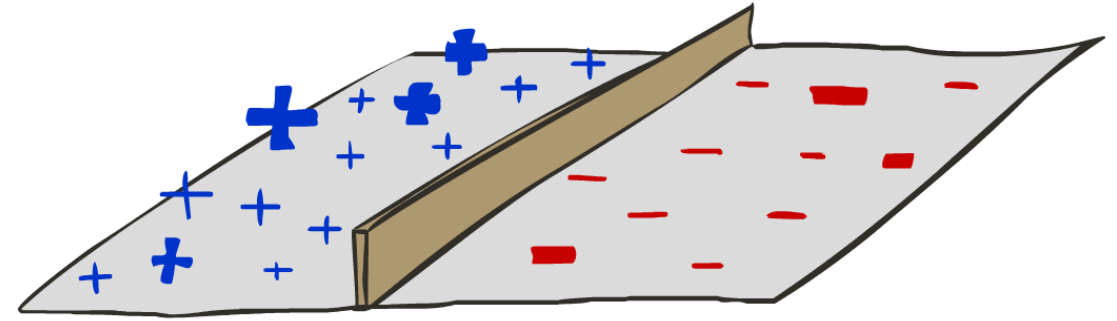
*Dot product $w \cdot f$ positive
means the positive class*

Decision Rules



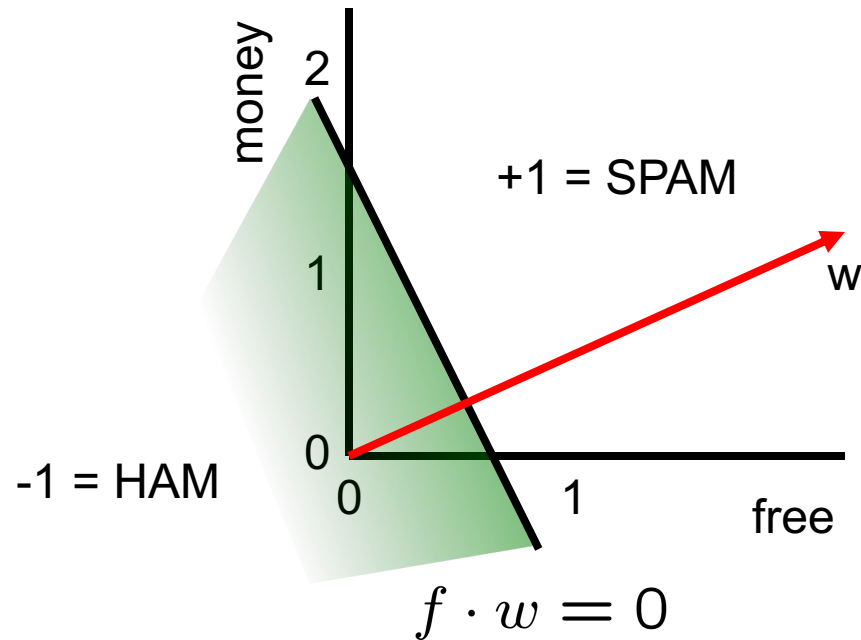
Binary Decision Rule

- In the space of feature vectors
 - Examples are points
 - Any weight vector defines a hyperplane
 - One side corresponds to $Y=+1$
 - Other corresponds to $Y=-1$



w

BIAS	:	-3
free	:	4
money	:	2
...		

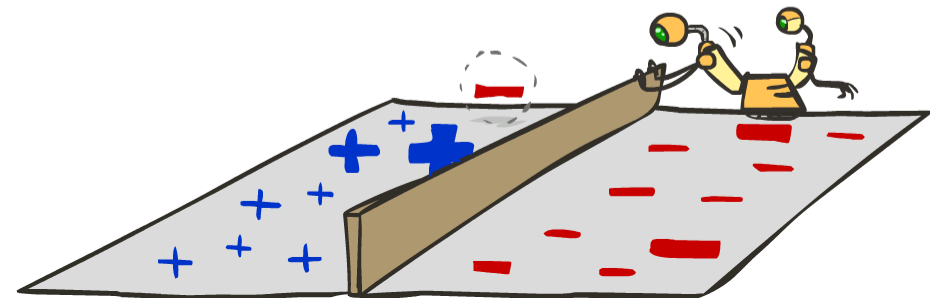
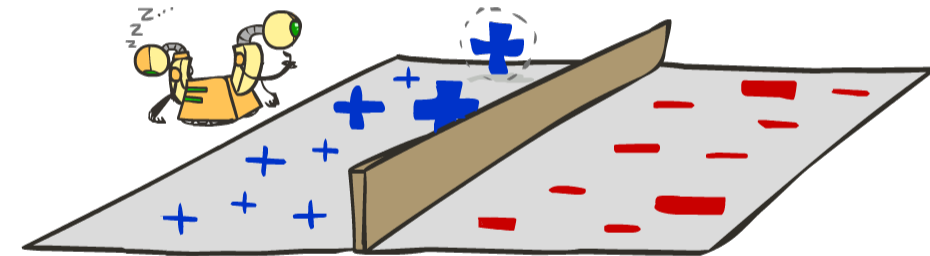
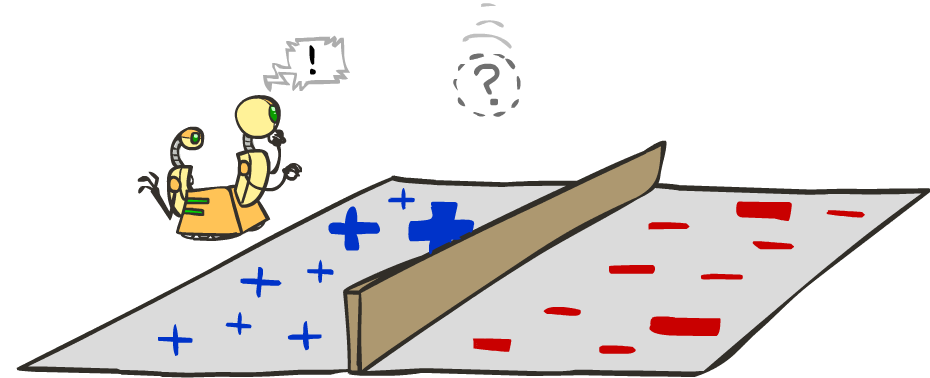


Weight Updates



Learning: Binary Perceptron

- Start with weights = 0
- For each training instance:
 - Classify with current weights
- If correct (i.e., $y=y^*$), no change!
- If wrong: adjust the weight vector



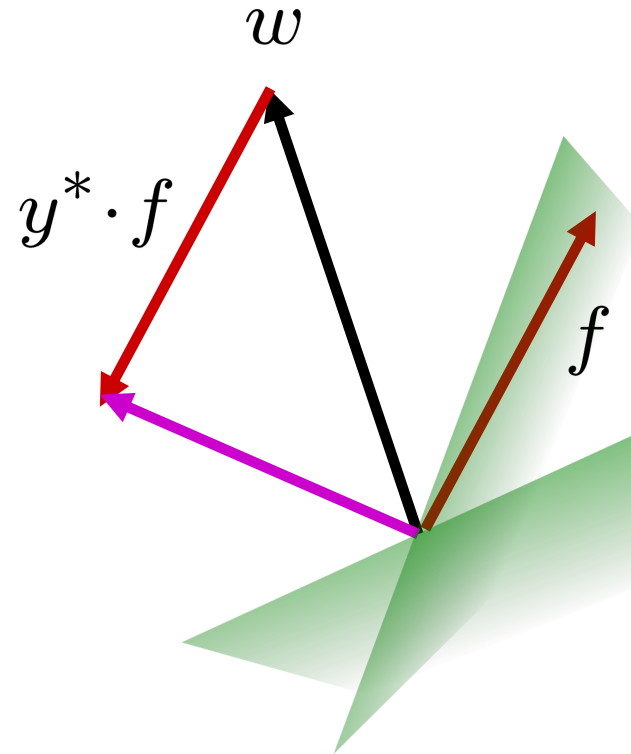
Learning: Binary Perceptron

- Start with weights = 0
- For each training instance:
 - Classify with current weights

$$y = \begin{cases} +1 & \text{if } w \cdot f(x) \geq 0 \\ -1 & \text{if } w \cdot f(x) < 0 \end{cases}$$

- If correct (i.e., $y=y^*$), no change!
- If wrong: adjust the weight vector by adding or subtracting the feature vector. Subtract if y^* is -1.

$$w = w + y^* \cdot f$$



Examples: Perceptron

- Separable Case

